解説
コンピュータ英語

日高哲郎・佐藤文博 編

共立出版株式会社

解説 コンピュータ英語

日高哲郎・佐藤文博 編

共立出版株式会社

解説 コンピュータ英語

日高哲郎・佐藤文博 編

共立出版株式会社

まえがき

現在は情報化時代と称されるように，多種多様の情報が存在しており，特定の知識を得ようとする場合，参考資料を容易に入手することができる。しかし，情報処理の分野では適当な和書がなく，洋書（英文記述の資料）に頼らざるを得ないことが多い。これは，情報処理技術の発展の経緯からいって当然のことであろう。さらに，当分野の国際性を考えると，今後はますますこの傾向が強まることが予想される。このため，情報処理の分野に問題意識のある者にとっては，レベルの差はあるにしても，英文資料を必要とし，かつ，英文の読解力が要求されてくる。

しかし，仮りに平易な英文で記述された資料であっても，英語の初心者にとって内容を把握することは非常に困難なことである。その理由は，第1に英語そのものに慣れ親んでいないため，英文を読むこと自体に抵抗があることである。第2に，特に情報処理の分野にいえることであるが，新しい専門用語，略語の知識がないと，前後の内容から類推しても，文意を的確に把握できないことがあげられる。第3に，情報処理の分野で慣用的に用いられている単語で，一般の辞書を参照しても和訳がむずかしく，また，和訳しない方が良い場合がある。

本書は，英語あるいは情報処理技術の初心者が，洋書を読む際の抵抗感を少しでも柔らげることを目的とすると同時に，情報処理技術者試験の受験者や，間接的に情報処理に携わる人々にも役立つように編集してあり，大学や，高専の初学年，各種学校のテキスト，副読本としても使用できるように配慮した。このため，比較的平易な文章を多くの分野から数多く取り上げ，見開き2頁を単位として，重要語句を説明した上で，情報処理の基礎知識も併せて吸収できるように構成した。

なお，内容は以下の通りで，大きく4つの部から構成されている。

第I部　基礎編　COBOL, FORTRAN, PL/I のプログラミングマニュアル，マイクロコンピュータの操作マニュアル，カタログ，雑誌記事など，日常参照することが比較的多いと考えられる基礎的な資料を取り上げた。

第II部　応用編　情報処理技術者試験に出題された問題のうち，平易でかつ内容的に特徴があり，コンピュータの知識が身につくと考えられる出題について問題文を取り上げ，解説を行うとともに解答を示した。

第III部　略語編　コンピュータ関係の資料に頻繁に使われる略語をアルファベット順に並べ，正式名称と内容の概略を示した。なお，一般用語，言語及び汎用パッケージ，団体名は分けて掲載し，使い易いように考慮した。

第IV部　用語編　JIS C 6230 情報処理用語対応英語をアルファベット順に掲載し，その内容を説明したもので，用語辞典として活用できる。

1982年8月10日

編者　日高哲郎
佐藤文博

目　　　次

第Ⅰ部　基　礎　編 …… *1*

第１章　プログラミングマニュアル …… *2*

〔１〕 COBOL …… *2*

（１） 言語の成り立ち …… *2*
（２） 適用される分野 …… *4*
（３） 言語の特徴 …… *6*
（４） 言語の構成と使用できる文字 …… *8*
（５） 予 約 語 …… *10*
（６） 定数（リテラル） …… *12*
（７） プログラムの構造 …… *14*
（８） レベル標識及びレベル番号 …… *16*

〔２〕 FORTRAN …… *18*

（９） 言語の特徴 …… *18*
（10） プログラムの実行手順 …… *20*
（11） ファンクションサブプログラム …… *22*
（12） リスト指示 …… *24*
（13） 入出力ルーチン—１— …… *26*
（14） 入出力ルーチン—２— …… *28*
（15） 診断機能 …… *30*

〔３〕 PL/I …… *32*

（16） プログラムの構造 …… *32*
（17） データの特徴と種類 …… *34*
（18） 省略時解釈 …… *36*
（19） 記憶域の割振り …… *38*
（20） 式 …… *40*
（21） 使用できる文字—１— …… *42*

(22) 使用できる文字—2— …… 44
(23) プログラムの構造 …… 46
(24) グループとブロック …… 48

第2章　オペレーションマニュアル …… 50
(25) 操　　作 …… 50
(26) 状態（モード） …… 52
(27) プログラミング …… 54
(28) 命令文 …… 56
(29) 電池の交換 …… 58

第3章　カタログ …… 60
(30) 周辺装置のオプション …… 60
(31) ソフトウェアの照会 …… 62
(32) システムプログラム …… 64
(33) ワークステーション …… 66
(34) アプリケーション …… 68

第4章　ニュース，論文，報告書 …… 70
(35) 分散型システムの概念—1— …… 70
(36) 分散型システムの概念—2— …… 72
(37) 利用の手引 …… 74
(38) 通信用端末装置 …… 76
(39) ソフトウェア管理簿 …… 78
(40) 端末装置とネットワーク …… 80
(41) 次世代のコンピュータ …… 82
(42) マイクロコンピュータ市場 …… 84
(43) コンピュータの世代 …… 86
(44) コンピュータシステム一般 …… 88
(45) 入力システムの動向 …… 90

(46)　技術革新 …… 92
(47)　印刷装置の動向 …… 94
(48)　第4世代のコンピュータ …… 96

第II部　応　用　編 …… 99

第5章　ハードウェア …… 100
(49)　計算機用語—1— …… 100
(50)　中央処理装置 (CPU) …… 102
(51)　命令 (INSTRUCTION) …… 104
(52)　中央処理装置 (CPU) と印刷装置 …… 106
(53)　タイムシェアリングシステム …… 108

第6章　ソフトウェア …… 110
(54)　プリプロセッサ …… 110
(55)　言語一般 …… 112
(56)　プログラミング …… 114
(57)　在庫管理モデル …… 116
(58)　プログラムの変更 …… 118
(59)　システムソフトウェア …… 120
(60)　ファイル編成 …… 122
(61)　システム設計 …… 124
(62)　シミュレーション …… 126

第7章　そ　の　他 …… 128
(63)　計算機用語—2— …… 128
(64)　数の表現 …… 130
(65)　正数，負数の表現 …… 132
(66)　計算機用語—3— …… 134
(67)　計算機システム …… 136

(68)　計算機用語—4— ……138
(69)　プログラム及びサブルーチン ……140
(70)　マイクロコンピュータの動向 ……142
(71)　パーソナルコンピュータ ……144
(72)　システム開発一般 ……146
(73)　データベース管理システム（DBMS） ……148

第III部　略語辞書編 ……151

〔1〕　一般用語 ……152
〔2〕　言語及び汎用パッケージ ……158
〔3〕　団体及び機関名 ……160

第IV部　JIS情報処理用語編 ……161

和文索引 ……198
英文索引 ……201

第Ｉ部　基礎編

基礎編では，英文解釈のトレーニングと情報処理の基礎知識を得ることを目的とした。収録した分野は，プログラミング言語，機械の操作，カタログ，一般記事及び雑誌である。

プログラミング言語では，COBOL，FORTRAN 及び PL/Iのプログラミング言語マニュアルから英文を選び，各言語の文法について解説し，雑誌やマニュアルを読むときに役立つよう配慮した。

機械の操作では，マイクロコンピュータの操作説明書から英文を抜粋し，このような説明書を読む場合の一般的パターンについて解説した。

カタログでは，マイクロコンピュータ，汎用コンピュータの周辺機器から英文を選んだ。カタログは一般に紋切り型の文が多いので，その意味のとらえ方，慣例的な表現などについて説明した。

一般記事・雑誌は，範囲が広く，また統一的な表現方法がないため，コンピュータの知識の吸収を主眼に説明した。

第1章　プログラミングマニュアル

〔1〕 COBOL

(1) 言語の成り立ち

COBOL (COmmon Business Oriented Language) is a programming language, similar to English, that is used for commercial data processing. It was developed by the Conference On DAta SYstems Languages (CODASYL).

The standard of the language is American National Standard COBOL X 3. 23-1968 as approved by the American National Standards Institute (ANSI). American National Standard COBOL is compatible with, and identical to, the international standard ISO/R 1989-1972 Programming Language Cobol.

(出典 : IBM OS Full American National Standard COBOL, © IBM Corp. 1970)

【重要語句】

- similar to～　　～と類似した
- is used for～　　～のために使われる
- was developed by～　　～によって開発された
- (as) approved by～　　～によって認められた（ものとして）
- is compatible with, and identical to～　　～と矛盾せず，同一である

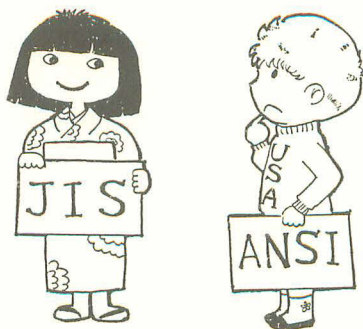
【解説】 コボルのマニュアルのはしがき (preface) の最初の部分に述べられている文章である。英文の読解という意味では非常に平易な構文であるため問題はなかろう。むしろここでは COBOL の開発主体である CODASYL や標準化機構について，常識として覚えておきたい。

COBOL　COmmon Business Oriented Language の略で，共通事務用言語の意味。プログラミング言語のうちの代表的なコンパイラ言語として位置づけられる。主に事務計算用に使用されるもので，コンピュータの知識がない

一般の人にも記述し易いように英語に近い表現形式をとっている。このため コンパイラ言語の中では現在最も多く使用されている。CODASYL を中心として 1960 年に開発された。

CODASYL Conference On DAta SYstem Language の略で，データシステム言語会議の意味。アメリカ政府，コンピュータメーカ及び使用者会議団体の三者から構成されている会議で，COBOL 言語の開発の推進母体となった。

ANSI American National Standards Institute の略で米国規格協会の意味。以前は ASA，USASI と称していたもので，日本の JIS と同様に標準規格の制定・認可を行う機関である。ANSI は多くの国際規格の幹事団体となっている。

ISO International Organization for Standardization の略で，国際標準化機構の意味。商品及びサービスの国際的交換を円滑にするための標準化の促進をはかる機関である。コンピュータに関しても，用語，プログラム言語，情報伝送上の諸特性などの標準化をはかっている。

〔メ モ〕

(2) 適用される分野

In 1959, a group of computer professionals, representing the U. S. Government, manufacturers, universities, and users, formed the Conference On DAta SYstems Language (CODASYL). At the first meeting, the conference agreed upon the development of a common language for the programming of commercial problems. The proposed language would be capable of continuous change and development, it would be problem-oriented and machine-independent, and it would use a syntax closely resembling English, avoiding the use of special symbols as much as possible. The COmmon Business Oriented Language (COBOL) which resulted met most of these requirements.

As its name implies, COBOL is especially efficient in the processing of business problems. Such problems involve relatively little algebraic or logical processing; instead, they usually manipulate large files of similar records in a relatively simple way. Thus, COBOL emphasizes the description and handling of data items and input/output records.

In the years since 1959, COBOL has undergone considerable refinement and standardization. Now, an extensive subset for a standard COBOL has been approved by ANSI (the American National Standards Institute), an industry-wide association of computer manufacturers and users;this standard is called American National Standard COBOL (formerly known as USA Standard COBOL).

(出典:IBM OS Full American National Standard COBOL, © IBM Corp. 1970)

【重要語句】

- be capable of～　～が可能な
- as much as possible　可能な限り
- manipulate　操作する
- subset　一部分，サブセット

【解説】 COBOL 言語の適用分野に関して述べている。一般的に COBOL は事務処理用言語であるといわれているが，この事務処理用という内容に2つ

の表現が用いられている。1つは「…a common language for the programming of commercial problems」であり，他方は，「business problems」である。直訳すると，前者は「商業上の問題をプログラミングするための共通言語」，後者は「事務の問題」であるが，いずれも，商業問題と訳しても内容に影響はない。また，プログラミングの負担を軽減するため，COBOL言語は「problem-oriented and machine-independent」であるとされている。これは「問題に即しかつ機械に依存しない」という意味であり，プログラマが自分が使用する計算機のことを意識しないで，プログラミングができることを意味している。

［メ　モ］

(3) 言語の特徴

COBOL is one of a group of high-level computer languages. Such languages are problem-oriented and relatively machine-independent, freeing the programmer from many of the machine-oriented restrictions of assembler language, and allowing him to concentrate instead upon the logical aspects of his problem.

COBOL looks and reads much like ordinary business English. The programmer can use English words and conventional arithmetic symbols to direct and control the complicated operations of the computer. The following are typical COBOL sentences:

```
ADD DIVIDENDS TO INCOME.
MULTIPLY UNIT-PRICE BY STOCK-ON-HAND
    GIVING STOCK-VALUE.
IF STOCK-ON-HAND IS LESS THAN ORDER-POINT
    MOVE ITEM-CODE TO REORDER-CODE.
```

Such COBOL sentences are easily understandable, but they must be translated into machine language——the internal instruction codes ——before they can actually be used.

(出典：IBM OS Full American National Standard COBOL, © IBM Corp. 1970)

【解説】 COBOL言語の特徴の説明を行っている。COBOLは，日常我々が使用する英文の形式（ほとんどが命令形）でプログラムを記述できる。そのために，アセンブラ言語のように使用する計算機のことを意識せずに，プログラムを作成することができる。このようなことをmachine-independent（機械から独立している）という。しかし計算機の内部コードは2進数であるため，作成されたプログラムを何らかの形で計算機の内部コードに変換する必要がある。この働きをするのがcompiler（コンパイラ）であり，それゆえ，COBOL言語はコンパイラ言語といわれることもある。コンパイラ言語には，他に，FORTRAN, PL/I, ALGOL などがある。一方，high-level computer languageという表現は，「高水準計算機用言語」という意味であるが，一般的には，「ハイレベルランゲージ」と呼ばれている。事実上は，COBOL, FORTRANなど，コンパイラ言語も含む呼び名であるが，機械の影響を受けず，問題を解く

ために作成された言語である。ハイレベルランゲージでは，使用する計算機のハードウェアの仕様については意識することなく，その問題を解くための論理を形成することに集中できるため，より高度な処理が可能となる。しかし，machine-oriented な言語（機械語あるいはアセンブラ言語のように，ハードウェアなど計算機の仕様を意識する必要のある言語）に比較して，きめ細かな処理ができない場合がある。また，処理速度が多少低下することもある。

ただし，machine-independent であるといっても，実際には COBOL 言語は，計算機によって多少記述の方法が違う場合がある。

［メ　モ］

(4) 言語の構成と使用できる文字

The COBOL language is so structured that the programmer can write his individual problem program within a framework of words that have particular meaning to the COBOL compiler. The result is the performance of a standard action on specific units of data. For example, in a COBOL statement such as MOVE NET-SALES TO CURRENT-MONTH, the words MOVE and TO indicate standard actions to the COBOL compiler. NET-SALES and CURRENT-MONTH are programmer-defined words which refer to particular units of data being processed by his problem program.

COBOL CHARACTER SET

The complete character set for COBOL consists of the following 51 characters:

Character	Meaning
0, 1, ……, 9	digit
A, B, ……, Z	letter
	space
+	plus sign
−	minus sign (hyphen)
*	asterisk
/	stroke (virgule, slash)
=	equal sign
$	currency sign
,	comma
;	semicolon
.	period (decimal point)
”	quotation mark
(	left parenthesis
)	right parenthesis
>	“greater than” symbol
<	“less than” symbol

(出典:IBM OS Full American National Standard COBOL, © IBM Corp. 1970)

【解説】 COBOL言語の構成とその文字セットに関して述べている。結論からいうと，プログラマはデータの定義だけをすれば，あとは，COBOLの定められた標準的な表現方法でプログラムを記述できるということが述べられている。この文章に記述されているように，「MOVE A TO B.」と書けば，ある定まった標準的な動作を行う。これは，領域AからBにその内容が転送されることを意味する。プログラマは，領域間の転送を行う場合，AとBに相当する領域だけ定義し，あとは，MOVE TO という標準的な表現を用いれば良いのである。

〔メ モ〕

(5) 予 約 語

Reserved words exist for syntactical purposes and must not appear as user-defined words. However, reserved words may appear as non-numeric literals, i.e., a reserved word may be enclosed in quotation marks. When used in this manner, they do not take on the meaning of reserved words and violate no syntactical rules.

There are three types of reserved words:

1. Key Words. A key word is a word whose presence is required in a COBOL entry. Such words are upper case and underlined in the formats given in this publication.

   Key words are of three types:

   a. Verbs such as ADD, READ, and ENTER.

   b. Required words, which appear in statement and entry formats, such as the word TO in the ADD statement.

   c. Words that have a specific functional meaning, such as ZERO, NEGATIVE, SECTION, TALLY, etc.

2. Optional Words. Within each format, upper case words that are not underlined are called optional words because they may appear at the user's option. The presence or absence of each optional word in the source program does not alter the compiler's translation. Misspelling of an optional word, or its replacement by another word of any kind, is not allowed.

(出典：IBM OS Full American National Standard COBOL, © IBM Corp. 1970)

【解説】 この文は，予約語に関する説明の一部を抜粋したものである。予約語とは，プログラマが勝手に用いてはならない語であり，定められた方法でのみ用いることができる。ただし，非数字リテラルとして引用符で囲んで用いる場合はさしつかえない。キーワードの説明において，ZERO, NEGATIVE, SECTION, TALLY などは特定の機能をもつとあるが，これらの語は，定数の確保（ZERO），領域の状態の判断（NEGATIVE），プログラム領域の構造の定義（SECTION），特殊な処理のためのカウンタ（TALLY）などに用いられるものである。すなわち，これらの語は，プログラマが勝手に用いてはい

けないのであり，定められた目的のために定められた方法でしか用いられないということである。また，文字定数として用いることは良いとあるが，これは，例えば，印刷の見出しなどに TALLY と印刷したいような場合は引用符(″)で囲って定数（リテラル）として〝TALLY″のように用いることはかまわないということである。

[メ　モ]

(6) 定数（リテラル）

A literal is a string of characters whose value is determined by the set of characters of which the literal is composed. Every literal belongs to one of two categories, numeric and non-numeric.

NUMERIC LITERALS: There are two types of numeric literals: fixed-point and floating-point.

A fixed-point numeric literal is defined as a string of characters chosen from the digits 0 through 9, the plus sign, the minus sign, and the decimal point. Every fixed-point numeric literal:

1. Must contain from 1 through 18 digits.
2. Must not contain more than one sign character. If a sign is used, it must appear as the leftmost character of the literal. If the literal is unsigned, the literal is positive.
3. Must not contain more than one decimal point. The decimal point is treated as an assumed decimal point, and may appear anywhere in the literal except as the rightmost character. If the literal contains no decimal point, the literal is an integer.

（出典：IBM OS Full American National Standard COBOL, © IBM Corp. 1970）

【重要語句】

- fixed-point numeric literal　固定小数点リテラル
- floating-point numeric literal　浮動小数点リテラル
- non-numeric literal　非数字リテラル

【解説】 COBOL で扱うリテラルのうち，数字リテラルの固定小数点リテラルについて説明した文である。リテラルとは，プログラム・ステートメントに内容をそのままの形で書かれた定数のことであり，アセンブラまたはコンパイラによって適当な記憶装置（リテラルプールという）に集められ，他の定数と同じように参照される。このとき，数値データは，計算機の内部形式に変換される。例えば，コーディング上で 13.2 と書くと，内部的には小数点は無視され，132 が格納される。そして，小数点は仮想小数点として取り扱われる。仮想小数点とは，記憶装置上には小数点は存在しないが，プログラムの指定によ

って，適当な位置に小数点があるものと仮定することである。具体例を下表に示す。

記憶装置	プログラムの指定	説　　明
12345	123.45	3と4の間に小数点があると想定する
12345	1.2345	1と2　〃
12345	1234.5	4と5　〃

また，英文には説明はないが，浮動小数点リテラル（floating-point リテラル）では，指数部と仮数部に分離されて格納される。

〔メ　モ〕

(7) プログラムの構造

Every COBOL source program is divided into four divisions. Each division must be placed in its proper sequence, and each must begin with a division header.

The four divisions, listed in sequence, and their functions are:

- IDENTIFICATION DIVISION, which names the program.
- ENVIRONMENT DIVISION, which indicates the machine equipment and equipment features to be used in the program.
- DATA DIVISION, which defines the nature and characteristics of data to be processed.
- PROCEDURE DIVISION, which consists of statements directing the processing of data in a specified manner at execution time.

Note: In all formats within this publication, the required clauses and optional clauses (when written) must appear in the sequence given in the format, unless the associated rules explicitly state otherwise.

(出典：IBM OS Full American National Standard COBOL, © IBM Corp. 1970)

【重要語句】

- in sequence　順序どおりに，次々と
- consist of　～から成る
- unless　もし～しなければ

【解説】 COBOL言語の構造について述べた文である。4つのディビジョンがあり，それらは全訳に示す順番に並んでおり，各々のディビジョンは，それぞれ機能をもっている。このディビジョンは，一般の日本語のマニュアルでは，「部」と訳されており，各々，見出し部，環境部，データ部，手続き部と表現されているが，ディビジョンと英語で表現することも多い。これらのディビジョンは，部見出し (division header) で始まり，必要な内容が記載される。部見出しとは，IDENTIFICATION DIVISION, ENVIRONMENT DIVISION, DATA DIVISION, PROCEDURE DIVISION である。以下に各ディビジョンの役割について説明する。

IDENTIFICATION DIVISION (見出し DIVISION の内容)	} 見出し DIVISION	・見出しディビジョン　プログラムの名称，作成日付などを記述する。
ENVIRONMENT DIVISION (環境 DIVISION の記述)	} 環境 DIVISION	・環境ディビジョン　プログラムで使用する計算機に関して指示する部であり，計算機によって書き方が違うことが多い。
DATA DIVISION (データ DIVISION の記述)	} データ DIVISION	・データディビジョン　プログラムが処理するファイルの論理的性質（長さ，データの形式など）及び作業領域のデータの長さ，形式などを指示する。
PROCEDURE DIVISION (手続き DIVISION の記述)	} 手続き DIVISION	・手続きディビジョン　プログラムと呼ばれる部分で，実際に行う処理を，処理順序に従って指示する。

プログラムの構造

〔メ　モ〕

(8) レベル標識及びレベル番号

In those Data Division entries that begin with a level indicator, the level indicator begins in Area A, followed in Area B by its associated file-name and appropriate descriptive information.

In those data description entries that begin with a level number 01 or 77, the level number begins in Area A, followed in Area B by its associated data-name and appropriate descriptive information.

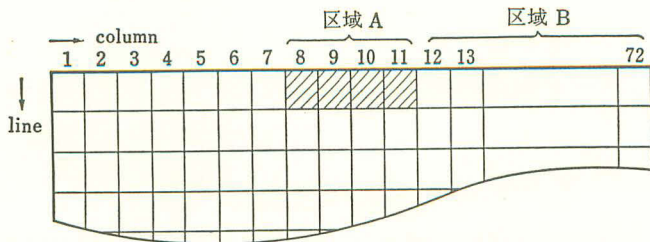
In those data description entries that begin with level numbers 02 through 49, 66, or 88, the level number may begin anywhere in Area A or Area B, followed in Area B by its associated data-name and descriptive information.

(出典：IBM OS Full American National Standard COBOL, © IBM Corp. 1970)

【重要語句】

▩ appropriate　適当な，特有の

▩ follow　～の次に来る，続く

【解説】 COBOL 言語のデータディビジョンにおけるデータの定義方法について述べている。ここでは，区域A，区域Bの意味を理解しておかないと，意味がつかめない。この英文では，区域Aを「Area A」，同様に区域Bを「Area B」と表現しているが，「margin A」，「margin B」と表現し，各々「Aマージン」，「Bマージン」ということも多い。これらの区域A，区域Bとは，COBOL プログラムを記述する用紙（コーディング用紙：coding sheet という）の位置のことであり，区域Aは，8カラム（column）から11カラム（column），区域Bは，12カラムから72カラムの間をいう。

コーディング用紙の例

前図で示すようなコーディング用紙に，1行（line）につき1ステートメントずつ記入していく。横方向には1ますにつき1文字ずつ記入してゆく。例えば，レベル番号01は区域Aから書くとあるから次のいずれの書き方でも良い。

→column

8	9	10	11	
0	1			(関連するデータの特性についての情報の記述)
	0	1		(　　〃　　)

↓ line

次に，レベル標識，レベル番号について説明する。**レベル標識**は，一般的には，レベルインディケータと英語そのままで呼ばれることが多い。これは，レベル標識に続く記述の内容を指示するものであり，「FD」(File Description)，「SD」(Sort Description) がある。「FD」の場合では，次に続く記述はファイルの性質を指示することを示し，「SD」は，以降の記述が分類の内容を指示していることを示している。一方，**レベル番号**とは，COBOL データ構造を階層構造で示すときに用いられる番号であり，上位のレベルにあるデータほど若い番号がつけられる。最上位のレベルは01を与え，以下02から順次付与し49まで与えることができる。88，66は特殊な用途，例えば，データの内容の判断，領域の再定義など，特殊な指示を行うときに用いる。

［メ　モ］

〔2〕 FORTRAN

(9) 言語の特徴

FORTRAN (FORmula TRANslator) is a programming language that's especially useful for applications involving mathematical computations and other manipulations of numeric data. It's particularly suited, therefore, to scientific and engineering applications.

FORTRAN looks and reads much like mathematical equations, so that programmers can use conventional mathematical symbols and constructions to control computer operations. FORTRAN is problem-oriented and relatively machine-independent. This frees FORTRAN programmers from annoying machine restrictions and lets them concentrate on the logical aspects of their data processing problems.

Compared with machine-oriented languages, FORTRAN gives you easy program development, decreased debugging effort, and improved overall data processing efficiency.

(出典：IBM VS FORTRAN Compiler and Library General Information, © IBM Corp. 1970)

【重要語句】

- useful for　～の役に立つ
- therefore　それゆえ
- like　類似した
- so that　～それで

【解説】 この英文はFORTRANの特徴を概念的に述べたものであり，特に難解な部分はないが，コンピュータ関係の英文にしばしば使われる用語が用いられているので，以下に簡単に説明する。

application（アプリケーション）「適用」とか「使用」あるいは拡大解釈して「適用業務」という訳語はあるが，日本語にしてしまうと，その内容がとらえにくくなってしまう。アプリケーションという単語自体で内容を理解し

ておいた方が良い。アプリケーションを簡単にいうと,「与えられた問題を解くために作成されたプログラム」のことである。また,FORTRAN の特徴として scientific and engineering applications という表現が用いられているが,直訳すれば全訳で示した通りであり,科学技術計算と訳して良い。

problem-oriented（プロブレムオリエンテッド）と machine-oriented（マシンオリエンテッド） problem-oriented とは,問題向きということであり,プログラミングに際して,使用する計算機の仕様などに原則として制限を受けないことを意味する。従って汎用性がある。一方 machine-oriented とは,機械向きということであり,使用する機械の仕様に応じてプログラミングを行う必要がある。従って汎用性はないが,きめ細かな処理が可能となる。前者の例として,FORTRAN, COBOL, PL/I などがあり,後者の例としては,アセンブリ言語がある。

computer（コンピュータ）と machine（マシン） コンピュータ関係の英文では,特殊な場合を除き,ほとんどコンピュータと訳してさしつかえない。ただし,ソフトウェアに関する記述の場合は computer,ハードウェアに関する記述の場合は machine としている場合が多い。

debugging（デバッギング） プログラムのテストを行うことをデバッグ（debug）あるいはデバッギング（debugging）という。日本語では,「虫取り」という訳語をあてている。

〔メ モ〕

(10) プログラムの実行手順

> Source programs written in FORTRAN consist of statements written to conform to FORTRAN rules.
>
> A FORTRAN compiler then analyzes the source program statements and translates them into machine language, which is suitable for execution on a computer system; FORTRAN compilers usually also produce other output to assist the programmer in debugging the source program.
>
> A FORTRAN compiler executes under control of an operating system, which provides input/output and other services.
>
> (出典：IBM VS FORTRAN Compiler and Library General Information, © IBM Corp. 1970)

【解説】 FORTRAN コンパイラの機能について述べた英文である。全訳で説明したように，ほとんどが，情報処理用語そのものを理解しておけば良いものである。

source program（ソースプログラム） 原始プログラムと訳す。一般的には，「プログラミング言語によって書かれた計算機用プログラム」のことをいう。すなわち，FORTRAN, COBOL, PL/I などで書かれたプログラムのことをいう。原始プログラムは，この英文からも分るようにコンパイラによって機械語に翻訳しなければならない。

compiler（コンパイラ） プログラミング言語は，人間が理解しやすい記号で書かれているが，コンピュータは，機械語（machine language）しか理解できないため，プログラミング言語を機械語に変換する必要がある。この働きを行うのがコンパイラである。直訳すると「編集者」というような意味になるが，現実的には，機械語への変換のことを「翻訳」としている。しかしコンパイラを「翻訳者」と訳すことはほとんどない。

input/output（インプット/アウトプット） コンピュータの重要な機能の1つである入出力を表現するのに，このような表現（間をスラッシュで区切る）が頻繁に用いられる。日本語では「入出力」をあてはめている。入力媒体としては，磁気テープ，磁気ディスク，カードなどが挙げられる。一方，出力

媒体としては，磁気テープ，磁気ディスクなどがある。このような装置以外に印刷装置も重要な媒体である。印刷装置の場合，どちらかというと print の方が適切であると考えられるが，output とすることも多い。本文の「原始プログラムをデバッグをする……作成します」という表現は，印刷と考えると分りやすい。

原始プログラムが翻訳されると，まず原始プログラムのリストと文法違反が印刷される。さらに文法違反をなくした後，プログラムを実行させると，もしエラーなどがあれば，その内容が印刷される。これがデバッグをするプログラマの手助けをするということである。

operating system（オペレーティングシステム） O.S. と省略して呼ばれることが多い。これは，計算機全体の使用効率を高めるため，各処理プログラム間のむだをなくし，操作員の介入を最小限にする目的で作られた制御プログラムの集合体である。

〔メ　モ〕

(11) ファンクションサブプログラム

Table 3 lists mathematical function subprograms and their entry names. Entry names starting with the letter D (except for DBLE and DIM) indicate double precision; starting with Q, extended precision(H Extended only); starting with C(except for CMPLX, CONJG, COS, COSH, COTAN), complex functions. Entry names starting with CD indicate complex double precision; starting with CQ, complex extended precision (H Extended only).

Table 3. Mathematical Function Subprograms

General Function	Entry Name
Natural and common logarithm	LOG, ALOG, DLOG, QLOG, CLOG, CDLOG, CQLOG, LOG10, ALOG10, DLOG10, QLOG10
Exponential	EXP, DEXP, QEXP, CEXP, CDEXP, CQEXP
Square Root	SQRT, DSQRT, QSQRT, CSQRT,CDSQRT, CQSQRT
Arcsine and arccosine	ASIN, ARSIN, DARSIN, QARSIN, ACOS, ARCOS, DARCOS, QARCOS
Arctangent	ATAN, DATAN, QATAN2
Sine and cosine	SIN, DSIN, QSIN, COS, DCOS, QCOS
Absolute value	IABS, ABS, DABS, QABS, CABS, CDABS, CQABS
Error function	ERF, DERF, QERF
Maximum and minimum values	MAX, MAX0, AMAX0, MAX1, AMAX1, DMAX1, QMAX1, MIN, MIN0, AMIN0, MIN1, AMIN1, DMIN1, QMIN1
Truncation	AINT, DINT, QINT, INT, IDINT, IQINT
Obtain real part of a COMPLEX argument	REAL, DREAL, QREAL
Obtain imaginary part of a COMPLEX argument	IMAG, AIMAG, DIMAG, QIMAG
Precision increase	DBLE, QEXT, QEXTD
Express two REAL arguments in complex form	CMPLX, DCMPLX, QCMPLX
Obtain conjugate of a COMPLEX argument	CONJG, DCONJG, QCONJG

(出典：IBM FORTRAN Program Products for OS and the CMS Component VM/370, © IBM Corp. 1974)

【解説】 FORTRAN で用いることのできる関数についての説明である。文章は平易であるが，辞書で調べても和訳が困難な単語及び慣例的に覚えておいた方が良いものがいくつかあるので，以下に説明する。

Table 表を指すときに用いられる。Table 3 とあれば表3ということである。また，図を指す場合は Figure が用いられる。

mathematical function subprograms mathematical は「数学の」という意味であるが，一般的には mathematical を和訳上では反映させず，「関数サブプログラム」という。

entry name 一般的には，「エントリネーム」あるいは「エントリポイント」ということが多い。これは，関数サブプログラムを用いるときに用いる名前であり，例えば，値Xの正弦（SIN）値をAに求めたい場合

A=SIN（X）

というように定義することになっており，このときの SIN が入口名である。

precision 精度ということであり，有効桁数を示す。double precision とは，倍精度といわれ，一般の実定数のほぼ2倍の精度をもつ。extended precision はさらに精度の高いものである。

〔メ モ〕

(12) リスト指示

> List-directed processing of READ, WRITE, PUNCH, or PRINT statements is specified by replacing the FORMAT statement label with an asterisk. No FORMAT statement is used. Data to be read at execution of a statement may be entered without regard to column boundaries. Individual data items may not be split between cards or lines, except for complex items and literals in quotation marks. Additional information on list-directed input and output is provided below :
>
> (出典 : IBM FORTRAN Program Products for OS and the CMS Component of VM/370, © IBM Corp. 1974)

【解説】 この英文はリスト指示の概要について述べた文である。FORTRANでは，入出力様式を FORMAT 文で定義することによって書式を制御する。しかしこのリスト指示を用いると FORMAT 文が不要であることを，この文章では述べている。そのためアスタリスク「*」を用いる。

次に，この英文では，enter と input が用いられている。これらの単語は，英文を読む上では，ほとんど同義と考えて良い。どちらも「入力」と考えて良い。厳密にいうと，entryは，「外部から，処理プログラムの要求に応じて，端末装置などからデータを与える」ことをいう。同じような意味の語に，to be read at execution of statement……がある。正確に訳せば全訳で示すような表現になるが，簡単に「入力データ」と訳しても，意味は十分に伝達できる。

さらに，慣用的に使う語がいくつかあるので，以下に説明する。

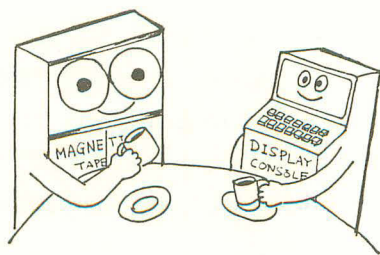
boundaries (boundary) 境界ということであるが，「バウンダリ」と一般的にいわれる。この英文では column. boundaries と用いられている。直訳すると，「列の境界」であるが，具体的には，次のようなことである。

例えば，複数のデータをカードから入力する場合，一般的にはデータの記述すべき位置 (column) が決められている。これを boundary という。すなわち，データの境目というような意味になる。この英文では，without regard to column boundaries とあるから，データの境界を意識しなくても良いということになる。他の表現では，free format といういい方がされる

こともある。また，may not be split between cards or lines という表現は，個々のデータを端末装置から入力する場合，複数枚のカードあるいは複数行にわたって記述してはいけないという意味になる。

except for　日本文でよく見られる「～を除いて」という表現を行うときに用いられる。良く出てくるので覚えておいた方が良い。

additional infomation on～　直訳すれば「～に関する付加的な情報は……」ということになるが，慣用的に，「～に関して詳細は……」という意味としてとらえておけば良い。

complex item and literals in quotation marks　complex は数学用語で複素数のことである。literals は，計算機用語としては名詞として用いられ，「文字定数」という意味になる。文字定数は，次に示すように引用符（quotation marks）で囲む約束になっているので，このような表現が用いられる。

〔メ　モ〕

(13) 入出力ルーチン—1—

VS FORTRAN provides input/output routines for FORTRAN input /output statements; these routines coordinate the interface with the system input/output routines.

Routines are provided for the following statements:

OPEN statement—makes a file available to the FORTRAN program

READ statement—retrieves a record from an external or internal unit

WRITE statement—transmits a record to an external or internal unit

ENDFILE statement—writes an end-of-file record on an external sequential file

CLOSE statement—releases a file processed by a FORTRAN program

(出典：VS FORTRAN Compiler and Library General Information, © IBM Corp.1970)

【解説】 マニュアルなどでは，このように紋切り型の文章が頻繁に出てくる。このような場合，前後の文脈で判断できないため，各単語の意味を十分に理解しておかないと意味がつかめない。なお，最初の3行の英文では，VS FORTRAN が主語になっている。VS FORTRAN は，IBM 社の FORTRAN Compiler の名称であるが，このように計算機関係では処理系（この場合コンパイラ）が主語となる場合が多くみられる。

routine 「型にはまった仕事」という意味から派生して，計算機における入出力処理のようにある定まった処理を行うために，システムが用意した一連の命令群である。これらによって，プログラマは，複雑な処理を簡単な命令だけですませることができる。

interface この語も適当な訳語がなく，インタフェースという表現が一般的に用いられている。これは厳密にいうと，「情報信号の授受を行うように接続された2つ以上の装置間において，その接続の境界として考える仮想的な面」のことを指す。良く使われる表現に「インタフェースが同じ」があり，

これは，信号の授受の規格が同じであることを意味する。すなわち，接続が容易であるということになる。インタフェースが違うと，接続に際しては，装置間に，ソフトウェア，ハードウェア両面に対しての対策が必要となる。

unit 一般的には，「単位」，「構成単位」という意味であるが，特に計算機のハードウェア関係では，「装置」とするのが一般的である。

external or internal unit 外部装置または内部装置ということである。**外部装置**とは，磁気テープ装置，磁気ディスク装置などの計算機に接続されている外部の補助記憶装置のことを指す。**内部装置**とは，計算機システムがもっている装置のことであり，一般的には，システム入力装置，システム出力装置が該当する。

end-of-file record ファイルの終りを示す印と考えて良い。なお，sequential file については，「(14) 入出力ルーチン―2―」で説明する。

〔メ モ〕

(14) 入出力ルーチン—2—

BACKSPACE statement—backspaces a sequential file one record
REWIND statement—causes the next READ or WRITE statement to process the first record in the file
INQUIRE statement—requests information about a file
FORMAT statement—specifies the structure of FORTRAN records
STOP statement—suspends execution of the program permanently
PAUSE statement—suspends execution of the program temporarily
DEFINE FILE statement(old FORTRAN only)—makes a direct access file available to the FORTRAN program
FIND statement (old FORTRAN only)—locates a direct file record

(出典：VS FORTRAN Compiler and Library General Information, © IBM Corp.1970)

【解説】

backspace 「戻る」と解釈しておけば良い。この英文では，「順次編成ファイルを1レコード戻す」と解釈できる。他に端末装置などでデータ入力時，1文字前を修正したいようなとき，タイプライタキーの BACKSPACE (BS) キーを押下することがある。このような場合も，backspace という。

sequential file, direct file ファイルの編成を表現する語である。sequential file は順次編成ファイルまたはシーケンシャルファイルといわれ，ファイルの先頭から順次レコードを取り出すファイルのことをいう。direct file は，直接編成ファイルまたはダイレクトファイルといわれ，ファイル中の任意のレコードを直接取り出すことが可能なファイルのことである。direct file は，random file (乱編成ファイル，ランダムファイル) といわれることもある。また sequential, random の両者を兼ね備えたファイルとして indexed sequential file (索引順次編成ファイル，インデックスシーケンシャルファイル) がある。

なお，これらのファイルをアクセス (レコードを取り出すこと，access) する方法から見て，sequential access method (SAM), direct access method (DAM), indexed sequential access method (ISAM) という表現が用いられることもある。なお ( ) 内は，省略形である。

REWIND　英文を直訳すると逆に分りにくくなってしまうが，このREWINDとは，磁気テープで考えると分りやすい。REWINDとは，みた目には磁気テープを巻き戻す処理のことと考えておけば良い。

STOP, PAUSE　STOPとPAUSEの違いは，permanentlyかtemporarilyかの違いである。永続的に停止とは，プログラムの実行が終了することを意味し，処理結果が，印刷装置などに出力されてくる状態をいう。一時的に停止とは，一旦プログラムは停止するが何らかの操作を与える（通常は，コンソールから再実行を指示する）か，一定時間後に，プログラムが再実行される状態をいう。

old FORTRAN　文脈からはIBM社のVS FORTRANの前のFORTRANコンパイラを指しているものと推定される。いずれにせよ，名称であるからあえて和訳する必要はない。

〔メ モ〕

(15) 診断機能

Diagnostic information issued by the compiler and by the execution-time library is more precise and more informative than ever before.

Compile-Time Messages from the compiler give the following information:

- The compiler phase number
- The message number
- The module name
- The severity level of the message:

  I—for Informational (notes to the programmer concerning compilation conditions)

  W—for Warning (possible program or machine error)

  E—for Error (errors exist that the compiler has attempted to correct; correct execution is possible)

  S—for Severe Error (errors exist that the compiler cannot correct; the statement in error is not processed further and subsequent coding errors within this statement can't be found or diagnosed during this compilation)

  U—for Unrecoverable Error (errors exist that cause the compiler to stop processing before the compilation is completed)
- The instruction number at which the error occurred
- The message text, telling what conditions caused the error

(出典：VS FORTRAN Compiler and Library : General Information, © IBM Corp. 1970)

【重要語句】

▩ ever before　それ以前に　　▩ following　以下に（述べる）

【解説】 FORTRAN プログラムの翻訳及び実行時に出力される各種情報のうち，翻訳時に出力される情報について述べた英文である。

diagnostic プログラムを翻訳した際，文法違反があると，該当個所を示す番号，理由などが印刷される。プログラマは，この内容をみてプログラムを修正し文法違反をなくす。ちょうど，コンパイラがプログラムを診断するよう

なことになるので diagnostic（診断の）という表現が用いられる。

severity　エラーの致命度を示す表現に用いられる。この英文では5段階あると説明されている。一般的にも，表現の差はあるが，このように1文字で表現される。Wは，例えば，桁あふれのように，データが正しく処理されない可能性はあるが，そのことをプログラマが認識していれば問題のないエラーを指す。一般的には，**ワーニング**と呼ばれる。

message text　直訳すると，「メッセージ本体」ということになる。コンパイラの出力する情報は種々あるので，あえて message text という表現を用いているようである。すなわち，番号や記号ではなく，文章が出力されることを意味している。

〔メ　モ〕

〔3〕 PL/I

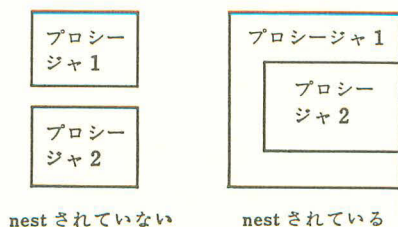
(16) プログラムの構造

A PL/I program consists of one or more blocks of statements called procedures. A procedure may be thought of as a subroutine. Procedures may invoke other procedures, and these procedures or subroutines may be either compiled separately, or nested within the calling procedure and compiled with it. Each procedure may contain declarations that define names and control allocation of storage.

The rules defining the use of procedures, communication between procedures, the meanings of names, and allocation of storage are fundamental to the proper understanding of PL/I at any level but the most elementary. These rules give the programmer considerable control over the degree of interaction between subroutines. They permit flexible communication and storage allocation, at the same time allowing the definition of names and allocation of storage for private use within a procedure.

(出典：OS PL/I Checkout and Optimizing Compilers : Language Reference Manual, © IBM Corp. 1976)

【解説】 PL/I のプログラム構成に関して述べた英文であるが，単語のもつ内容についてある程度予備知識がないと分りにくい。

まず，「PL/I は，**プロシージャ**と呼ばれる文（ステートメント）の集合から成る」とある。これは，そのあとに「プロシージャは，サブルーチンとみなされる」とあるので，概念的には，PL/I は，サブルーチンの集合であると解釈できる。従って一般的にもいえることであるが，サブルーチン単位に別々に翻訳できると説明されている。なおプロシージャ及びステートメントは一般的に各々「手続き」,「文」と訳される。また，あるプロシージャをネスト式に組み込む場合には，そのプロシージャと一緒にコンパイルできると次に説明があるが，これが PL/I の特徴の１つでもある。**ネスト式**（nest）は,「入れ子」と訳す。例えば,「プロシージャが入れ子になっている」といういい方が用いられる。これは，概念的には，次図のような状態のことをいう。

次に，「PL/Iを正しく理解するためには」とある。これは，データの名前や形式だけでなく，プロシージャの用途，プロシージャ間の連絡（サブルーチンを呼び出すときの引数の受け渡しなどのことを指す），領域の割当制御（PL/Iでは，領域を**静的**（static）に確保する場合と**動的**（dynamic）に確保する場合がある）などに関する規則に精通する必要があると説明されているが，初歩的な段階ではPL/Iの省略時解釈（プログラマが指示しない場合，PL/Iが自動的にデータの特性などを解釈する）にまかせて十分である。

〔メ モ〕

(17) データの特徴と種類

The characteristic of PL/I that most contributes to the range of applications for which it can be used is the variety of data types that can be represented and manipulated. PL/I deals with arithmetic data, string data (bit and character), and program control data, such as labels. Arithmetic data may be represented in a variety of ways; it can be binary or decimal, fixed-point or floating-point, real or complex, and its precision may be specified.

PL/I provides features to perform arithmetic operations, comparisons, and operations and functions for assembling, scanning, and subdividing strings.

The compiler must be able to determine, for every name used in a program, the complete set of attributes associated with that name. The programmer may specify these attributes explicitly by means of a DECLARE statement; the compiler may determine all or some of the attributes by context; or a partial or complete set of attributes may be assumed by default. The programmer can specify which attributes are to be applied by default, or he can allow the compiler to determine them.

(出典：OS PL/I Checkout and Optimizing Compilers; Language Reference Manual, © IBM Corp. 1976)

【重要語句】

▩ such as　～のような，例えば～

【解説】 PL/I の扱うデータの特徴と種類について述べた文章である。PL/I で取り扱うことのできるデータの特性（形式や長さなどのデータの性質のことで，**属性**という）は，プログラマが定義しない場合は，コンパイラによって自動的に決定されると説明されている。また，データの演算に関してストリング演算にその特徴がある。character string といえば，文字の列のことをいい，bit string といえばビットの列のことをいう。これらのストリングに関する演算には全訳で示したように組立て，走査，細分がある。ストリングの組立てとは，左に示すように2つの列を結合して新しい列を作ることである。ストリン

'ABCD' ‖ 'EFGH'

↓（組立て）

新しい文字列　'ABCDEFGH'

ストリングの組立て

グの走査とは，列の内容を調べることである。ストリングの細分とは，列の一部を取り出して新しい列を作ることである。

ストリング演算の代表的な関数であるSUBSTR組込み関数について簡単に説明しておく。領域Aの内容が'123456'という文字ストリングであり，SUBSTR (A, X, Y) とすると，これは，領域AのX桁目からY桁を取り出すことを示す。従って，B=SUBSTR (A, 1, 2) とすれば，領域Bは，'12'という文字ストリングが格納されることになる。また，IF SUBSTR (A, 2, 1) = '3' とすれば，領域Aの2桁目から1文字は'2'であるからこの比較（検査）の結果は偽となる。

〔メ　モ〕

(18) 省略時解釈

An important feature of PL/I is its default philosophy. If all the attributes associated with a name, or all the options permitted in a statement, are not specified by the programmer, attributes or options will be assigned by the compiler. This default action has two main consequences. First, it reduces the amount of declaration and other program writing required; second, it makes it possible to teach and use subsets of the language for which the programmer need not know all possible alternatives, or even that alternatives exist.

The default attributes assumed by the compiler are the standard default attributes of the PL/I language and the implementation precision defaults. However, the programmer can override these by use of the DEFAULT statement.

The compiler optionally produces an attribute listing which contains the identifiers used in a PL/I source program and a complete list of the attributes specified either by explicit, contextual, or implicit declarations, or by application of default rules. The programmer can use this listing to check that these attributes are consistent with his intentions.

(出典：OS PL/I Checkout and Optimizing Compilers：Language Reference Manual, © IBM Corp. 1976)

【解説】 PL/I の扱うデータの属性の省略時解釈について述べた文である。PL/I では種々のデータを取り扱うことができるが，その属性（attribute）を与える方法として，explicit（明に），contextual（文脈上），implicit（暗に），DEFAULT statement（DEFAULT 文）がある。「明に」とは，プログラムにおいてその属性を明確に与えることをいう。この表現は，一般には，単語そのものを用いて「エクスプレシットに」ということの方が多い。「文脈上」とは，プログラムの前後の記述によってコンパイラが自動的に解釈し，属性を与える場合をいう。「暗に」とは，プログラムでは，属性の記述をせず，コンパイラにその属性の決定をまかせることをいう。この表現も，「明に」の場合と同様単語そのものを用いて「インプリシットに」ということの方が多い。

「DEFAULT 文」の場合は，全訳でも述べたように，プログラムで，DEFAULT 文を用いて属性を与える方法である。

次に identifier(s) について説明する。これは，**識別名**と訳されるが，「アイデンティファイア」と呼ばれることも多い。identifier は，PL/I 言語の最小単位である言語用文字構成（アルファベット，数字，特殊文字どな）から選んだいくつかの文字，例えば変数とかファイル名などを他のものと識別するために与える文字列のことをいう。従って，データ名，ファイル名，ラベルなどと考えれば良い。

〔メ　モ〕

(19) 記憶域の割振り

PL/I goes beyond most other languages in the flexibility of storage allocation that it provides. Dynamic storage allocation is comparatively difficult for an assembler language programmer to handle for himself; yet it is automatically provided in PL/I. There are four different storage classes: AUTOMATIC, STATIC, CONTROLLED, and BASED. In general, the default storage class in PL/I is AUTOMATIC. This class of storage is allocated whenever the block in which the variables are declared is activated. At that time the bounds of arrays and the lengths of strings are calculated. AUTOMATIC storage is freed and is available for re-use whenever control leaves the block in which the storage is allocated.

Storage may also be declared STATIC, in which case it is allocated when the program is loaded; it may be declared CONTROLLED, in which case it is explicitly controlled by the programmer with ALLOCATE and FREE statements, independent of the invocation of blocks; or it may be declared BASED, which gives the programmer an even higher degree of control.

(出典：OS PL/I Checkout and Optimizing Compilers: Language Reference Manual, © IBM Corp. 1976)

【重要語句】

▩ goes beyond～　　を越える

▩ storage allocation　　記憶領域割振り

【解説】 PL/I で用いる各種の記憶領域の割振り方法について述べた文である。記憶領域の種別として，「AUTOMATIC」，「STATIC」，「CONTROLLED」，「BASED」がある。「AUTOMATIC」が指定されると，そのブロックが起動されたときのみ領域が割り当てられる。そしてこのブロックの処理が終了するとこの領域は解放される。このように，プログラム実行時に動的（dynamic）に割り振られる領域を AUTOMATIC 属性をもつ領域と呼ぶ。動的な割振りとは，次に説明する静的（static）と対になる用語であるが，最初から割り振られるのではなく，ブロックが呼び出されるつど割り振られることをい

う。一方，静的な領域を STATIC 属性をもつ領域と呼ぶ。この領域はプログラム起動時に割り振られる領域で，プログラムの全ての処理が終了するまで解放されることはない。また，CONTROLLED，BASED は，ブロックの起動とは別個に必要なつど領域を割り振る場合に指示する。

さらに，PL/I における領域の割振りは，アセンブラ言語と比較すると非常に簡単であると述べられている。このため，プログラマは，対象となるアプリケーションの要求（処理速度を向上させるのか，記憶領域を少なくするのか，プログラミングの経済性を考慮するのか）に応じて種々のプログラミングを行うことができるが，動的割振りを行うと処理速度が低下する。これは当然のことである。なぜならばプログラムの実行中に領域を確保すると，そのための時間が必要となるからである。そのため，プログラムの**スループット**（thruput：総処理時間）は低下する。

〔メ モ〕

(20) 式

Calculations in PL/I are specified by expressions. An expression has a meaning in PL/I that is similar to that of elementary algebra. For example:

A+B*C

This specifies multiplication of the value of B by the value of C and adding the value of A to the result. PL/I places few restrictions on the kinds of data that can be used in an expression. For example, it is conceivable, though unlikely, that A could be a floating-point number, B a fixed-point number, and C a character string.

When such mixed expressions are specified, the operands will be converted so that the operation can be evaluated meaningfully. Note, however, that the rules for conversion must be considered carefully; converted data may not have the same value as the original. And, of course, any conversion increases execution time.

The results of the evaluation of expressions are assigned to variables by means of the assignment statement. An example of an assignment statement is:

X=A+B*C;

This means: evaluate the expression on the right and store the result in X. If the attributes of X differ from the attributes of the result of the expression, conversion will again be performed.

(出典：OS PL/I Checkout and Optimizing Compilers: Language Reference Manual, © IBM Corp. 1976)

【重要語句】

- for example～ 例えば～
- so that～ ～するように
- evaluation 評価

【解説】 PL/I においては，算術演算の対象となるデータの種類（属性のこと）は原則として制限がない。しかし，属性が違う場合には，演算可能とするため，内部的に変換が行われる。このとき，変換誤差などのため，予期した結果が得られない場合があると説明されている。さらに変換のための時間が増え

る。また演算に関して operation という単語が用いられているが，この単語は，「操作，動作，作動」という意味もあるので和訳には注意が必要である。本文のように計算と訳す場合は日本語でよいが，操作という意味に用いるときは，「オペレーション」ということが多い。また，operand という単語は，適格な日本語がないため「オペランド」ということが多い。これは本来は「計算機の命令語（instruction）の構成において，データや，次の命令の貯えられているアドレス（address：番地）などを示す部分」のことである。本文のように式の場合は右辺の各項を operand と表現している。次に，evaluation について簡単に説明する。本文では計算を行うことを，evaluation（評価）としているが，PL/I の場合には，よく用いられる。これは，演算の結果に対して，属性の変換を行うことによるものである。

〔メ　モ〕

(21) 使用できる文字—1—

One of two character sets may be used to write a source program: either a 60-character set or a 48-character set. For a given external procedure, the choice between the two sets is optional. In practice, this choice will depend upon the available equipment.

60-CHARACTER SET

The 60-character set is composed of alphabetic characters, digits, and special characters.

There are 29 alphabetic characters beginning with the currency symbol ($), the number sign (#), and the commercial "at" sign (@), which precede the 26 letters of the English alphabet in the IBM System/360 collating sequence Extended Binary Coded Decimal Interchange Code (EBCDIC). For use with languages other than English, other characters may be substituted for $, #, and @.

There are ten digits. The decimal digits are the digits 0 through 9. A binary digit is either a 0 or a 1.

An alphameric character is either an alphabetic character or a digit.

(出典：OS PL/I Checkout and Optimizing Compilers: Language Reference Manual, © IBM Corp. 1976)

【重要語句】

- external procedure　外部処理手順
- is composed of～　～から成る
- currency symbol　通価単位
- "at" sign　単価記号
- collating sequence　照合順序
- alphameric　英数字（英字，数字との特殊文字の総称）

【解説】 まずPL/Iで使用可能な文字の種類（character set）には，60文字セット及び48文字セットの2種あることが説明され，次に60文字セットに使用できる文字のうち，アルファベット順文字及び数字の種類について説明されている。

この英文では，アルファベットに関して IBM 社特有の定義がなされている部分がある。それは，alphabetic character は，$, #, @ 及びA～Zの 26 文字，計29文字のことを指していることであり，A～Z は，English alphabet と定義されていることである。

これら29の文字については，大小関係が定められている。この大小関係の順序のことを**照合順序**（collating sequence：コレーティングシーケンス）と呼び，この照合順序を定めている体系のことを EBCDIC または EBCDIC コード体系という。EBCDICは，Extended Binary Coded Decimal Interchange Code の省略形で，拡張2進化10進コードと訳されているが，EBCDIC（エビセディック）と呼ぶことが多い。これは，各文字が計算機内に読み込まれたとき，内部コードに変換されるが，このコードが1バイト（8ビット）の2進数となる。この2進数の大小関係によって，文字の大小関係（collating sequence）が定まる。

また，英字と数字をまとめて表現するために，alphameric が使われているが，alphanumeric という単語もしばしば用いられる。

〔メ　モ〕

(22) 使用できる文字－2－

There are 21 special characters. They are as follows:

Name	Character
Blank	
Equal sign or assignment symbol	=
Plus sign	+
Minus sign	−
Asterisk or multiply symbol	*
Slash or divide symbol	/
Left parenthesis	(
Right parenthesis	)
Comma	,
Point or period	.
Single quotation mark or apostrophe	'
Percent symbol	%
Semicolon	;
Colon	:
"Not" symbol	¬
"And" symbol	&
"Or" symbol	|
"Greater than" symbol	>
"Less than" symbol	<
Break character	_
Question mark	?

Special characters are combined to create other symbols. For example, <= means "less than or equal to", ¬= means "not equal to". The combination ** denotes exponentiation (X**2 means X^2). Blanks are not permitted in such composite symbols.

The break character is the same as the typewriter underline character. It can be used in a name, such as GROSS_PAY, to improve readability.

(出典：OS PL/I Checkout and Optimizing Compilers: Language Reference Manual, © IBM Corp. 1976)

【解説】 文字のセットの説明のうち，特殊文字に関する説明を行っている。

以下に Break character と Assignment symbol について説明する。

Break character　英文の最後の部分で説明があるように，タイプライタのアンダーラインと同じ文字である。これは長いデータ名称などをつけるとき，途中にこの文字を入れることによって読み易くなる。

Assignment symbol　一般的に数式は A=B+C のように表現するが，数学では，B+C の結果とAが等しいことを意味する。しかし PL/I では，B+C の結果をAに代入することを意味する。このような式（PL/I では文とみなす）を Assignment statement（割当て文）といい，このとき用いられる＝を Assignment symbol と呼ぶ。

〔メ　モ〕

(23) プログラムの構造

A PL/I program is constructed from basic program elements called statements. There are two types of statements: simple and compound. These statements make up larger program elements called groups and blocks.

There are three types of simple statements: keyword, assignment, and null, each of which contains a statement body that is terminated by a semicolon.

A keyword statement has a keyword to indicate the function of the statement; the statement body is the remainder of the statement.

The assignment statement contains the assignment symbol (=) and does not have a keyword.

The null statement consists only of a semicolon and indicates no operation; the semicolon is the statement body.

Examples of simple statements are:

GO TO LOOP_3; (keyword statement) GO TO is a keyword; the blank between GO and TO is optional. The statement body is LOOP_3;

A=B+C; (assignment statement)

(出典：OS PL/I Checkout and Optimizing Compilers: Language Reference Manual, © IBM Corp. 1976)

【解説】 プログラムの構成要素である**ステートメント（文）**のうち，単純ステートメントに関して説明している。全訳から分るように，単純ステートメントには3種類あり，キーワードを含むステートメント（keyword statement），キーワードを含まないステートメント（割当て文，assignment statement）及び空白ステートメントがある。keyword statement は，先頭に**キーワード**（この英文例では，GO TO）があり，以下；（セミコロン）までのステートメント本体と呼ばれる語（英文例では LOOP_3;）が付加されている。キーワードとステートメント本体との間に1つ以上の空白が必要である。

割当てステートメントは，＝の右辺の結果を左辺に代入することを示してい

る。これは，いわゆる計算式であって，キーワードは含まれない。空白ステートメントは，；だけから成るステートメントで何もしない。なお，キーワードは，ステートメント識別語（statement identifier）と呼ぶこともある。また，ステートメントという表現は，「文」ということもある。英文では，取り上げなかったが，複合ステートメントについて簡単に説明する。例えば，

```
IF X>Y THEN X=Y+Z;
            ELSE X=Y-Z;
```

のように，ステートメント本体の一部として他のステートメントを含むものである。上記の例では，IF X>Y…X=Y−Z；の文の中に，X=Y+Z，X=Y−Z の2つのステートメントを含んでいる。

［メ　モ］

(24) グループとブロック

A group is a sequence of statements headed by a DO statement and terminated by a corresponding END statement. It is used for control purposes. A group also may be called a do-group.

A block is a sequence of statements that defines an area of a program. It is used to delimit the scope of a name and for control purposes. A program consists of one or more blocks. Every statement must appear within a block. There are two kinds of blocks: begin blocks and procedure blocks. A begin block is delimited by a BEGIN statement and an END statement. A procedure block is delimited by a PROCEDURE statement and an END statement. Every begin block must be contained within some procedure block.

Execution passes sequentially into and out of a begin block. However, a procedure block, except the first, must be invoked by execution of a statement in another block. The first procedure in a pro gram to be executed is invoked automatically by the operating system. This first procedure must be identified by specifying OPTIONS (MAIN) in the PROCEDURE statement.

(出典：OS PL/I Checkout and Optimizing Compilers : Language Reference Manual, © IBM Corp. 1976)

【重要語句】

▩ sequence 列　　▩ pass into～ ～に入る

【解説】 グループとブロックの概念について，説明した文である。**グループ**は，DO 文から始まり，対応する END で終り，DOグループとも呼ばれると説明されている。具体的には，左図のような状態である。

```
DO ……; ┐
  }文   ├ DOグループ
END;    ┘
```

ブロックは，概念的には次頁の図の通りである。

OPTIONS (MAIN) のあるプロシージャブロックは，オペレーティングシステムによって呼び出されるが，他のプロシージャブロックは，実行中に文によって呼び出される（プロシージャブロック2)。プロシージャブロックに対し

```
PROCEDURE…OPTIONS(MAIN);  ┐
 :                        │
BEGIN……;       ┐          │
 :             ├ 開始ブロック1
END……;         ┘          │
 :                        │
BEGIN……;       ┐          │
 :             ├ 開始ブロック2   ├ プロシージャブロック1
END……;         ┘          │
 :                        │
PROCEDURE;     ┐          │
 :             ├ プロシージャブロック2
END……;         ┘          │
END……;                    ┘
```

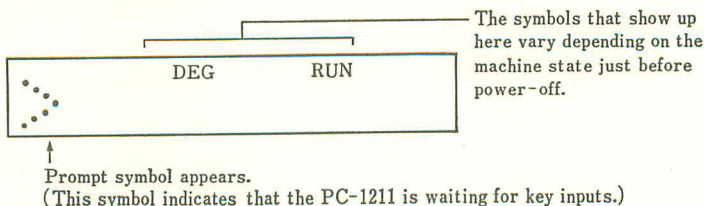
て**開始ブロック**（一般的にはビギンブロックということの方が多い）は，文の順序に従って実行される。左の例では，開始ブロック1，2の順に実行される。ただし，プロシージャブロックは実行されない。プロシージャブロック2の実行は，すでに説明したように，他のブロックからの呼出しによる。

〔メ　モ〕

第2章　オペレーションマニュアル

(25) 操　　作

First press the power switch ([ON]). The machine then gives the following display.

Prompt symbol appears.
(This symbol indicates that the PC-1211 is waiting for key inputs.)

The computer is supplied with programs and carries out computations according to the input programs. Besides, it can solve problems without being programmed, where they are too simple to require programming, if supplied with necessary data directly through its keyboard. The former process is called program calculation and the latter one manual calculation—this is also called DIRECT EXECUTION because of direct calculation.

1. Manual calculation

At first make the PC-1211 display the symbol "RUN" at the top of the display of the computer by pressing the [MODE] ([MODE] [MODE]) keys.

Now let's enter on practice.

RUN symbol
RUN

(1) Addition, Subtraction

Key in the following:

12 [+] 45.6 [−] 32.1 [+] 789 [−] 741 [+] 213

Note that as you key in the "3" in 213, you have exceeded the 24 character capacity of the display. At this point a unique feature called "rolling writer" becomes effective. As each additional step is entered, the display will roll to the left. The data rolled off the screen will be recorded up to 80 steps.

Now press [ENTER] (Do not press the [=] key). Your answer is 286.5

(出典：SHARP Pocket Computer MODEL PC-1211 Instruction Manual)

【重要語句】

▨ prompt　促進すること　　▨ former…latter～　前者の…後者の～

【解説】 display と screen のニュアンスの違いを説明する。display は，画面に表示されている内容を意味しており，screen は，装置自体のことをいう。また key in～ は元来，「キーボードを打鍵してデータを入力する」という意味であり，**キーイン**という表現で用いられることが多い。

〔メ モ〕

(26) 状態（モード）

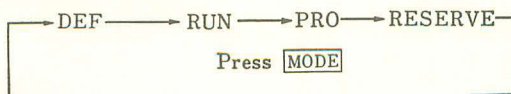
The PC-1211 has four modes: definable, run, program and reserve program, each mode can be set by pressing the [MODE] key located at the right side of the computer.

Definable mode (DEF): Puts the machine in the defined program execution mode. Perform defined program calculations in this mode.

Run mode (RUN): Puts the machine in the calculation execution mode. Perform program or manual calculations in this mode.

Program mode (PRO): Puts the machine in the program writing mode. Enter programs in this mode.

Reserved program mode(RESERVE): Puts the machine in the reserved program writing mode. Enter reserve programs in this mode.

The [MODE] key changes the mode.

(出典: SHARP Pocket Computer MODEL PC-1211 Instruction Manual)

【解説】 コンピュータの種々の状態とその働きについて述べている。**状態（モード）**は，一般的には**モード**ということの方が多い。また，definable, run, program, reserved program の各状態については，この計算機を用いる上でそれらの働きさえ理解できれば良いのであるから，あえて和訳する必要はない。各々，DEF モード，RUN モード，PRO モード，RESERVE モードというように理解しておけば良い。また，run と execution (execute) に関しては，両方とも「プログラムなどの実行」または「プログラムなどを実行する」という意味であり，英語の文法的な解釈はともかくとして，計算機関係の用語としては，同義語と考えて実用上はさしつかえはない。

〔メ モ〕

(27) プログラミング

The programming language the PC-1211 uses is BASIC language. The BASIC language, a dialogue language for scientific computations, is said to be the easiest to understand and use among a variety of programming languages and is widely used by people ranging from beginners to experts with programming.

1. What is a program calculation?

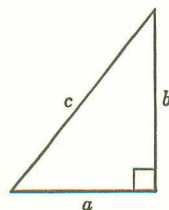
When computing for Pythagoras's theorem, for example, you must carry out the following operation.

Pythagoras's theorem

For a rectangular triangle, its three sides a, b and c have the following relations

$$c=\sqrt{a^2+b^2}$$

where c is the opposite side to the right angle.

The manual calculation requires the following operation. (when $a=3$ and $b=4$.)

Operation	Display	Note
Set to the RUN mode.		
[A] [=] 3 [ENTER]	3.	A is loaded with 3.
[B] [=] 4 [ENTER]	4.	B is loaded with 4.
[C] [=] [√‾] [(] [A] [*] [A]	C=√‾A*A_	
[+] [B] [*] [B] [)]	C=√‾A*A+B*B)_	
[ENTER]	5.	$\sqrt{A^2+B^2}$

A series of these key operations can be programmed as follows.

〔PROGRAM 1〕

Programming	Note
10: INPUT A, B	Input instruction
20: C=√‾(A*A+B*B)	Operation instruction
30 : PRINT C	Output instruction
40: END	End instruction

(出典 : SHARP Pocket Computer MODEL PC-1211 Instruction Manual)

【重要語句】

▨ rectangular triangle 直角三角形 ▨ right angle 直角

【解説】 手操作による計算は，実は BASIC 言語のプログラムを作り出しているのであるということが説明されている。そして，この BASIC 言語は，非常に使いやすく，初心者から熟練者に至るまで幅広く用いられている。なお，BASIC とは，Beginners All purpose Symbolic Instruction Code の略で，ダートマス大学の J.G. Kemeny らによって開発された FORTRAN に似た会話型プログラム言語である。

［メ モ］

(28) 命 令 文

In the descriptions given below, [variable], [numerical variable], [character variable] and 〈expression〉 have individually the following meanings.

[**Variable**] : General name of numerical and character variables.

[**Numerical variable**] : General name of fixed memories defined by A through Z and dimension memories defined in the form of A ( ).

[**Character variable**] : General name of fixed memories defined by A$ through Z$ and dimension memories defined in the form of A$ ( ).

〈**Expression**〉 : Operational expression composed of elements of 〈expression〉 shown on page 15, covering also [numerical variable].

1. LET statement

The variable name on the left is assigned the value of constant or expression on the right. The PC-1211 does not require LET except when it is defined for a statement following IF statement.

General form (1)	LET [Numerical variable]=〈Expression〉	
Example :	LET A=5*3	
Example :	LET A=123	Instruction to put 123 in A (LET can be omitted as in the following example.)
	A(30)=3*6	Instruction to put 18 in A (30).
General form (2)	LET [Character variable]="Character"	
Example :	LET Z$="BASIC"	

Characters between quotation marks are put in the character variable specified by the left side. When the length of a string of characters on the right side exceeds seven characters, however, the first seven characters alone are put in and the excess is discarded.

(出典 : SHARP Pocket Computer MODEL PC-1211 Instruction Manual)

【解説】 PC-1211 で使用するプログラム言語 BASIC の LET 文について

説明している。LET 文は，ある領域(全訳で示したように領域の定義の方法は，数字変数と文字変数とでは違う）に値を代入する文であり，このことを表現するのに「to put〜」が用いられている。これと似た表現には「to place」がある。ほとんど同義であるが，「文字変数」の場合ある領域に文字列を置くということからplace が用いられることが多い。次に，following, follows（次に）について説明する。通常は，follow (ing) があれば，段落を区切りその内容の詳細が説明されると考えてよいが，「statement following IF」(16 行目)のように「IF 文に続く」と訳されるような場合がときどきある。文脈から十分把握できるので，注意して訳す必要がある。

〔メ　モ〕

(29) 電池の交換

When the battery indicator is out, replace the designated mercury batteries.*

1. Turn off the computer.
2. Remove the screws from the back cover with a small screw driver (Fig. 1). (Please note that two kinds of screw are used.)
3. Replace the batteries. (Fig. 2)
4. Hook the tabs of the back cover into the slits of the computer proper. (Fig. 3)
5. Push the back cover in slightly while replacing the screws. Tighten the screws securely at this time.
6. Push the reset switch on the back cover to clear the computer. (Fig. 4) Use a ball-point pen to press the reset switch.
7. Press the [OFF] and [ON] keys to clear the computer. When the batteries are correctly installed ">DEG RUN·" will be displayed.

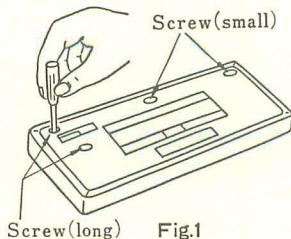
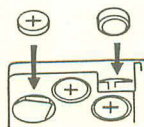
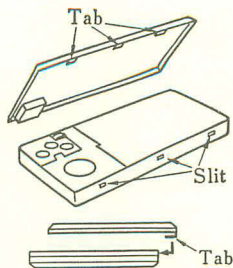
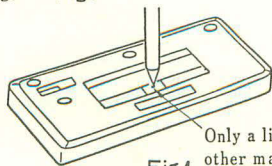
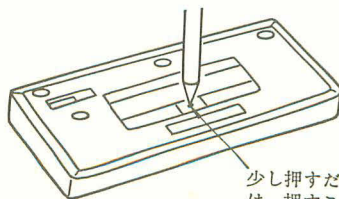



Fig.1 Fig.2 Fig.3 Fig.4

* Battery

● Mercury battery (Type MR 44)×4

Batteries may be obtained where you purchased your computer or at most retail outlets for calculators, watches, or cameras.

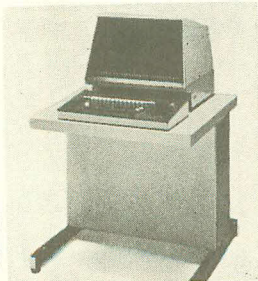
(出典：SHARP Pocket Computer MODEL PC-1211 Instruction Manual)

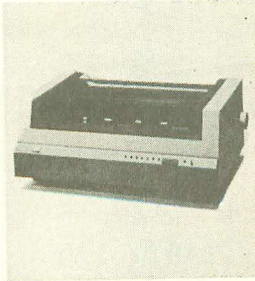
【解説】 電池交換の手順について述べたものである。電池の有無の状態が battery indicator に表示されると説明されている。battery indicator

は，「電池を指示するもの」すなわち，電池の状態を示すものと訳されるが，一般的には，**バッテリインディケータ**という。このバッテリインディケータが OFF（消える）になるとバッテリーが無くなったことを示す。

次に tab について説明しておく。本来は「止め金具」，「つけ札」といった意味であるが，タイプライタの「TAB」キーのように，ある一定の位置までスキップするような意味に使われることが多い。しかしこの場合は本来の意味で用いられており，計算機本体と裏ぶたを結合している金具のことを指している。

［メ モ］

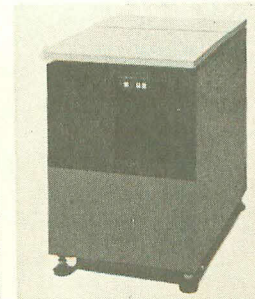
第3章　カタログ

(30)　周辺装置のオプション

WORK STATION

Terminal can be used independently within 3,000ft. from CPU in interactive mode. Transmission rate 1M bits/s.
Characters: 2,000 char. (80 char. x 25 lines).
Color: green (ruled lines and boxed fields available).
Numeric key pad and function keys are available. Detachable keyboard.

WORK STATION SERIAL PRINTER

Used for printing sales slips and lists at a work station.
Print Speed: 200 char./sec. Reciprocating print. Will print black and red. Ledgers and single slips may be used.

COLOR CHARACTER DISPLAY

Displays characters and graphics on high resolution tube in green, red and white.
Characters: 2,000 char. (80 char. x 25 lines).
Colors: green, red, white (ruled line display, possible).

LINEPRINTER -300/600 LPM

Processes monthly statements and reports in a short time by high speed print.

Print speed :300/600 l.p.m.

Print width: 132 char./line.

Characters: 64.

CARTRIDGE DISK DRIVE

Expands capacity indefinitely through replacement of removable cartridge disks.

Capacity : 10, 20 M bytes.

Mean access time: 57.5 m. sec.

DISK PACK DRIVE

Large capacity disk pack of 50 M bytes.

Capacity: 50 M bytes.

Mean access time: 38.3 m. sec.

(出典 : PROGRAMMED FOR GROWTH MITSUBISHI MODEL 8028/8038)

【解説】 省略形について以下に説明する。

M : メガ　10^6 倍のこと

LPM, l.p.m : Line/minute　1分間当りの印刷行数

x : アルファベットのxを乗算記号として使っている

ft. : feet の略　1 feet=12 inch≒30 cm

m. sec : ミリ秒　10^{-3} 秒のこと

〔メ モ〕

(31) ソフトウェアの照会

SOFTWARE.

We design *and* manufacture our own software, so you're assured of top performance throughout the Model 8018's operation. Our programs are written in easy-to-understand BUSINESS BASIC and come in a variety of applications. You may not require all our programs right away. But as your business grows, you can add the ones you need.

For immediate use, choose Mitsubishi's reasonably-priced turnkey programs. Or, if you prefer, ask us to modify our existing package. We're flexible. And so is our software.

APPLICATION PROGRAMS.

Inventory Management—reports, reviews and analyzes inventory to insure that the right product is in the right place, at the right time, for the right price.

Accounts Receivable—processes receivables efficiently and accurately to generate the necessary cash flow.

Sales Analysis—manages and analyzes each product, each salesperson and each customer to determine the most profitable channels to pursue.

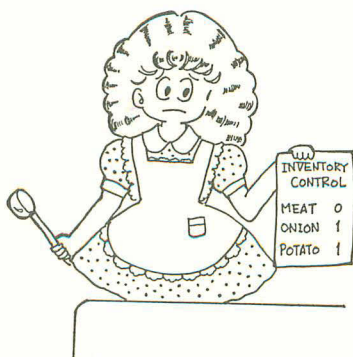
(出典：PROGRAMMED FOR GROWTH MITSUBISHI MODEL 8018)

【解説】 計算機に提供されているソフトウェアについて述べた部分である。最初の部分は，We から始まっているが，We をそのまま訳して行くと意味が分りにくい。この We は，Mitsubishi というソフトウェアの提供をしている会社を指しているのであるが，このような場合，software を主語にして理解していった方が良い。

アプリケーションの部分では，Inventory Management は在庫管理の意味であり，Inventory Control とする場合もある。また，Account Receivable は，売掛金処理と訳しておいたが，会計用語では，Account Receivable は「受取勘定」であり，売掛金は，Account Receivable-trade である。しかし，ここでは，業務名が並列に並べられているので，売掛処理，あるいは会計処理の方

が適切であろう。

また本文中に turnkey program とあるが，trunkey は，本来「牢番」,「看守」という意味であり，これが転じて，管理プログラムというようなニュアンスでとらえられている。しかし，実際には訳さなくても影響はない。

［メ モ］

(32) システムプログラム

SYSTEMS PROGRAMS.

Data conversion—a group of programs that ease the transition of data from your current system to the Melcom 8028/8038 system.

Supervisor—supervises all of the hardware and programs in operation.

Job management—manages job schedule and job source; executes processing programs.

Online control—manages work stations and terminals and provides easy access.

Data management—manages files (input/output devices and data) and provides access methods to files.

Library management—operates and manages libraries; provides access methods to libraries.

LANGUAGES PROCESSING.

BUSINESS BASIC—can be used for business-oriented processing, simplified language for both conversational and interpretive type.

RPG II—standard simplified language stressing compatibility with other computers.

PROGRESS II—enables dynamic processing of online and batch operation with abundant language functions.

(出典：PROGRAMMED FOR GROWTH MITSUBISHI MODEL 8028/8038)

【解説】 以下に翻訳上注意すべき単語を列挙する。

conversion と transition いずれも変換という意味で良いが，transition は，移行という意味が強い。従って，この英文でいえば，現在のシステムから Melcom 8028/8038 システムへの移行というニュアンスである。

supervisor 元来は管理者という意味であるが，コンピュータ関係では監視プログラムと訳す。これは，各種のプログラムを管理するプログラムを指すからである。「スーパーバイザ」とそのままいわれることが多い。

interpretive プログラムの命令を1つ1つ解釈しながら実行するものをインタープリタという。

dynamic　動的と訳す。動的とは，資源が必要なときに与えられたり，確保されたりすることをいう。例えば，プログラムの実行に必要な資源を確保するようなとき「dynamic に確保する」という。dynamic に対して static という表現がある。これは静的と訳す。static の場合には，上記の例でいえば，最初に必要な資源を確保しておくことをいう。

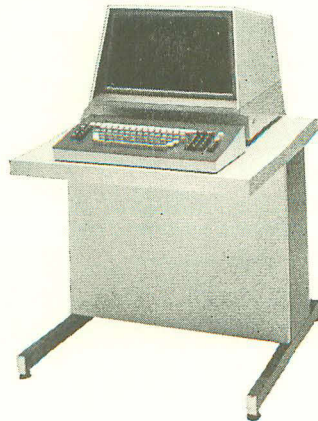
〔メ モ〕

(33) ワークステーション

Each work station is a terminal unit that is capable of starting programs in addition to data input. Works can be performed as required at any time at any place only by operating any one of the work stations located in various departments. So data processing can be performed quickly and efficiently.

Work stations which are indispensable for the multiple work system. The work station of MELCOM 80 Model 38 can be used privately by each department. It is of a design providing expandability to connect up to 32 work stations.

Functions of Work Station

1. Capable of displaying 2,000 characters on a large screen of 14 inches, now sales slips and graphs can be more clearly shown. Kanji characters also can be displayed (1,000 characters at maximum).
2. Display is shown in green, which provides easy observation and less fatigue.
3. Dual intensity (normal and pale tone) and blink functions, which are effective for emphasizing important part of the display and for informing input errors, are provided.
4. Ruled line display can be made for displaying the position and range of input characters.
5. The keyboard, containing ten-key for entry of numerics and function key enabling single touch operation makes operation easier.
6. Provides completed field control functions.
7. Hand OCR and other features can be added at option.

Private stand is optional.

（出典：未来を開発する三菱電機 ENGLISH VERSION）

【解説】以下に翻訳上注意すべき単語の意味について説明する。

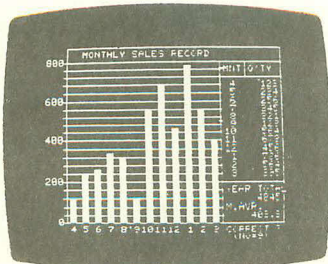
blink 画面上に表示された文字が点滅する機能。

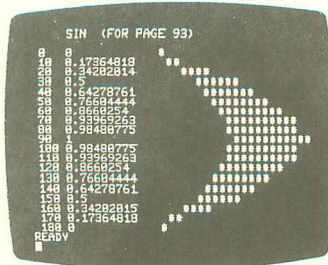
single touch 日本語ではワンタッチと訳される。1度の操作で何らかの処理ができるような操作。

ten-key 通常のタイプライタキーではなく，数字の0～9及び・（小数点）のみのキーボード。

〔メ モ〕

(34) アプリケーション

The MZ-80K has many applications including small business calculations, scientific computations, data banking, education, hobbies, etc.

1. Business

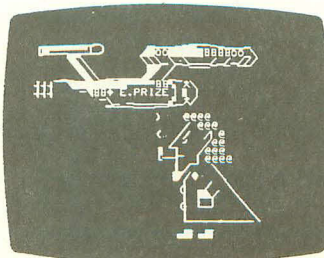
a) Marketing analysis b) Invoicing c) Costing d) Stock management e) Sales and purchase ledgers f) Payroll

2. Education aids

a) Mathematics
b) Science
c) Language

3. Scientific and industrial calculations

From the most, simple arithmetic calculations to the most complex computations.

4. Hobby/Home use

a) Computer games
b) Music
c) Home budgeting

BASIC is the standard conversational language programmed on the MZ-80K in tape mode. It can be expanded or augmented by use of a floppy disc. Moreover, Assembler Language, Machine Language and other computer languages can be applied by simply exchanging tapes.

PASCAL, FDOS will soon be available.

(出典：SHARP MZ-80 K Personal Computer System)

【解説】 このような書き方の場合，名詞の羅列が多いため，分りにくい用語が出てくるが，慣例的なものである。

costing 原価を計算するということから原価計算となる。

stock management 在庫管理であるが，inventory management とすることが多い。他に inventory control, stock control という表現もある。

payroll 給与支払簿ということから給与計算という意味で使われる。

language 言語教育でも国語教育としてもさしつかえないが，言語教育の方が幅広い意味になる。

home budgeting 家庭の予算ということから家計（管理）という訳が適当であろう。

［メ モ］

第4章　ニュース，論文，報告書

(35)　分散型システムの概念—1—

Honeywell Distributed Systems Environment (DSE) could more precisely be termed a set of concepts, rather than an architecture, as it does not require rigid adherence to specific procedures or equipment configurations. Although the network components which are offered by DSE would certainly seem to qualify it as an architecture, Honeywell prefers that it be called a concept.

In its first DSE announcement in January 1977, Honeywell termed DSE "an information processing concept incorporating a set of overall Honeywell design objectives." It is not surprising, therefore, that the typical DSE configuration consists of Honeywell Information System's two most successful products, the Level 66 mainframe and the Level 6 minicomputer.

DSE, as the name implies, embraces distributed processing: the placement of applications processing at the user site, which can generally be said to relieve the host of some of its processing burden, and reduce the cost of communications facilities, as line usage to a remote host is minimized.

(datapro, C 11-480-101 より引用)

【重要語句】

- term　呼ぶ，名づける
- adherence　固守，支持
- component　成分
- relieve　救う
- burden　負担，義務

【解説】 Honeywell 社の DSE についての紹介であるが，concept（**コンセプト**）か architecture（**アーキテクチャ**）かが議論の対象となっている。architecture は，構造，構成という意味であるが，一般にはアーキテクチャと呼

ばれることの方が多い。アーキテクチャは技術論であり，処理の手順や機器構成を具体的に示すことである。一方，concept は概念ということで，考え方を提供するということである。例えば，新しい機器を接続するようなとき，他社の製品よりも同一社の製品を用いる方が接続がスムーズにいくが，コンセプトの場合は，条件を満足するならば，他社の製品でも論理的にはスムーズに接続ができることを意味する。

次に**分散処理**（distributed processing）について説明する。分散処理とは，オンラインシステムで全ての処理を中央のコンピュータ(ホストコンピュータ)にさせるのではなく，一部の処理を端末装置にさせ，ホストコンピュータの負荷を軽減することである。例えば，簡単なエラーチェック，表示装置であれば，画面の編集等，端末装置特有の処理を行うことである。このようにすることにより，必要最小限のデータだけがホストコンピュータに伝送されるため，ホストコンピュータの負荷が軽減され，システム全体の稼動効率を高めることができる。さらに通信回線の使用量が減少するため，料金が安くなる。また，通信量が減るため，通信設備費用も減少する。

［メ　モ］

(36) 分散型システムの概念―2―

In addition to distributed processing, DSE supports distribution of data bases and data base management; a feature which the Honeywell user may or may not elect to employ, depending on the degree of remote processing required. DSE likewise supports the integration of a variety of systems and communications protocols into a single, efficient, distributed network which supports interactive, batch, time-sharing or a combination of these applications simultaneously.

Although Honeywell prefers that its equipment be implemented into a DSE network, provisions exist within DSE to support non-Honeywell terminals. Communications and remote data entry are likewise supported to non-Honeywell mainframes, namely IBM. Honeywell correctly recognizes that specific users needs will generally mandate a hybrid system comprised of DSE and non-DSE components, but cautions that the user will frequently experience a lesser degree of communications efficiency as the introduction of non-Honeywell equipment increases.

(datapro, C 11-480-101 より引用)

【重要語句】

- in addition to　…に加えて，さらに
- prefer　むしろ…を選ぶ
- mandate　…の統治を任命する
- comprise　含む
- namely　すなわち

【解説】 DSE がサポートする機能について説明している。DSE がサポートするものには種々あるが，そのために必要な機器は，Honeywell 社のものであってもなくても良い。ただ．Honeywell 社以外の機器が増えるに従って，通信能力が低下することは避けられないであろうと述べている。なお，**サポート**するという表現がこのような文献ではしばしば現われる。これは，提供するとか維持するという意味であり，ソフトウェアなど，人間以外を指す語（この場

合DSEというシステム）が主語となった場合に良く用いられ，「～ができる」という意味に解釈する。

以下，本文に出てくるデータベース及びプロトコルについて簡単に説明しておく。

data base　データの多目的利用を前提とした共用ファイル群であり，一般のデータ（ファイル）の構造がアプリケーションプログラムに左右されるのに対して，アプリケーションとは独立した構造をもつ。また，この共用ファイル群を管理する一連の機能を DBMS (Data Base Management System) または**データベース管理システム**という。

protocol　協定書という意味であるが，**プロトコル**と呼ぶことが多い。複数のプロセス間で授受されるデータの形式や授受のタイミングなどに関して定める規則のことをいう。本文で述べられている通信プロトコルの場合，伝送制御手順，誤り制御，情報の形式等が規定され，送信側と受信側はそのプロトコルに従って，情報を組み立てる。

〔メ　モ〕

(37) 利用の手引

Monthly you will receive major supplements and revisions to the publication to ensure continual reliability and timeliness of information. Also *every month* you will receive a copy of *Software News* that will keep you and your staff aware of important new developments in proprietary software.

Each supplement and each *Software News* includes easy-to-follow instructions for filing the new and revised reports. (You'll notice that all page numbers are arranged in straightforward alphanumeric sequence, although many reserved spaces have been left in the numbering system to facilitate later insertions.)

To ease the chore of filing reports even more, new reports as well as revised reports are numbered in such a way so as to permit filing behind all other similar software product reports within a major section of the volume. The index entries will be changed regularly to reflect all additions and deletions.

The *DATAPRO DIRECTORY OF SOFTWARE* is authored and edited by the technical staff of Datapro Research Corporation. The staff comprises experienced analysts, writers, and editors working under the direction of exceptionally qualified professionals in the field of EDP software.

(datapro, D 03-100-003 より引用)

【重要語句】

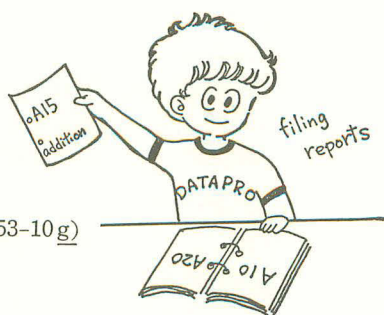
- supplement　補遺，増刊，付録
- revision　改訂版
- aware of～　～に気づいている
- facilitate　容易にする
- as well as　と同様に～も
- so as to～　～するように

【解説】 本文の出典である DATAPRO Report の前書きにある，いわば，「本書の用い方」といった部分である。DATAPRO Report は，一定期間ごとに追加修正削除が行われており，それらの変更に対処する方法についてユーザに説明している。またこれが DATAPRO Report の特徴でもある。

DATAPRO Report では，英字，数字で頁番号を構成することによってあ

き番号をつくるようにしている。そのため，ファイリングが容易であると述べている。なお，参考のために，DATAPRO Report の頁のつけ方を例示しておく。

(70c-533-10e → 70c-533-10f → 70c-53-10g)

［メ　モ］

(38) 通信用端末装置

The TermiNet 9610 Communications Terminal is a microprocessor-based terminal designed to support a variety of communications disciplines. Model 9610 is an upgraded replacement for the 9600, which is currently being phased out. The new model has all the capabilities of its predecessor, and contains an Intel 8080 A chip instead of the GE-designed microprocessor.

Program packages are currently available for emulation of the IBM 2780, IBM 3780, Honeywell GRTS-355, and GE's own Mark III time-sharing service.

Basic configuration consists of the 9610 Communications Controller and a TermiNet 340 Series line printer. The controller is designed for installation on the right side of the pedestal of the line printer, or can optionally be mounted inside the pedestal. A TermiNet 30 teleprinter can be added to the basic configuration where a keyboard is necessary to satisfy user requirements. Other TermiNet line printers, TermiNet teleprinters, or a compatible customer-supplied keyboard/display may be able to replace the TermiNet 340 or TermiNet 30 printers; contact GE for further information. (See Datapro's Report C 27-450-101 for a complete description of TermiNet teleprinters). GE also offers a 500-cpm card reader (produced by Peripheral Dynamics) and a magnetic tape terminal as attachments to the basic 9610.

(datapro, C 23-450-101 より引用)

【解説】 TermiNet 9610 端末装置の紹介記事である。マイクロプロセッサを用いることによって，従来の端末装置と比較して機能的に向上されており，さらに，装置自体の軽量，小型化がはかられている。以下にマイクロプロセッサに関して，簡単に説明しておく。

マイクロプロセッサとは，計算機の中央処理装置あるいは演算装置の部分を1基板上に構成したもので**集積回路**（IC : Integrated Circuit）の集合体である。また，IC が多数結合されていることから**高密度集積回路**（LSI : Large

Scale Integration)といわれる。この LSI は，前述したように，中央処理装置や演算装置に相当する部分が集積されているので，いわば，非常に小さな中央処理装置といえる。マイクロプロセッサに入出力制御回路や記憶装置を接続すると小型ながらコンピュータとしての機能を有する。これを一般には，**マイクロコンピュータ**（microcomputer）と呼んでいる。

マイクロコンピュータは，軽量小型であるため，本文で紹介されたような通信端末として以外にも多方面に利用されている。文字表示（Display）装置や POS (Point of Sales) 端末に組み込んで，編集機能や計算機能をもたせた，**知能端末**（インテリジェントターミナル：Intelligent terminal）がその代表的なものである。また，計算機以外の分野でも，電子式キャッシュレジスタ，数値制御（NC：Numerical Control），ゲーム電卓などにも使用されている。

［メモ］

(39) ソフトウェア管理簿

The DATAPRO DIRECTORY OF SOFTWARE represents the latest and most wide-ranging independent information service dedicated to comprehensive coverage of all varieties of proprietary software offered for sale to the computer user community. Designed as a tool for management and technical staffs at all levels, the service is compiled and edited to present in an organized and concise manner all the pertinent facts and evaluations you will need to better appreciate the fast growing software market.

The DIRECTORY is intended to be used in the following principal ways:

- As a *selection tool* for the prospective user to acquire products.
- As a *curent awareness vehicle* to aid the knowledgeable information specialist in keeping abreast of the dynamism in this industry segment.
- As a *planning guide* in determining the make-up of future systems enhancements and design.

As such, this publication is designed to be a convenient reference tool. For this reason, this User's Guide presents proven methods and techniques for obtaining maximum value from this service depending on the nature of your specific applicational needs.

(datapro, D 03-100-001 より引用)

【重要語句】

- dedicate to～　～にささげる
- pertinent　適切な
- make-up　性質，構成
- enhancement　向上，増強
- dynamism　物力論，力本説（あらゆる現象をすべて自然力の作用によるとする説）

【解説】 本文は，DATAPRO DIRECTORY OF SOFTWARE（データプロソフトウェア管理簿）の特徴を述べている部分であり，前書きに相当する

部分である。このような場合，表現が大げさになり，形容詞や副詞が連続的に現われるので，まず修飾語を取り払って和訳してみると良い。一般的には，これらの形容詞などが重要な意味をもつことは少ない。例えば，2行目の the latest and most wide-ranging independent は，「最も最近の，最も広範囲にわたった独自の」という意味であるが，これらをはずして和訳しても，要するに，DATAPRO DIRECTORY OF SOFTWARE が，ソフトウェアに関する情報をサービスしているのであるということは分る。なお，DIRECTORY という語は，住所，氏名録，人名簿という意味であり，一般的には，**ディレクトリ**と呼ばれている。これは，種々の情報の所在などを掲載してある管理簿といった意味である。

次に斜字体（*selection tool*, *current awareness vehicle*, *planning guide*）について説明する。全訳では，一応各単語のもつ個別の意味を考慮して訳しておいたが，このような場合，強調するというよりは，熟語としてとらえていく必要がある。このような熟語が実際にあるかどうかの議論は別として，執筆者は，熟語として意識している。従って，各々「選択手段」，「現状認識手段」，「設計手引」という方が良い。

［メ　モ］

(40) 端末装置とネットワーク

●NETWORKS AND ARCHITECTURES

The report on Honeywell's Distributed Systems Environment(DSE) has been completely revised and retitled. Honeywell's Distributed Systems Architecture(DSA), which is a subset of DSE, is a commitment by the company to the Open Systems Interconnection(OSI) developed by ISO.

●DISPLAY TERMINALS

You will find a revised report on the Beehive Alphanumeric Display Terminals.

There is a revised report on the General Terminal Corporation GT-100/GT-400 Display Terminals. General Terminal was formerly known as Infoton.

Also included is a revised report on the Teletype 4540 Display Terminals, which have been very well received in the marketplace as indicated by the User Reaction section of this report.

●TRANSMISSION FACILITIES

You will find a revised report on AT & T's Digital Data Service (DDS).

The reports on RCA Satellite Service and Switched Satellite Service have been revised.

Satellite Business Systems has been announcing customer contracts with some very large manufacturing and retail establishments. A revised report on their current service offerings is included.

There are revised reports on Telex, TWX message service, and Low Speed Channel Service of Western Union.

(datapro, "How to file" より引用)

【解説】 本文のように社名や機器名が羅列されている場合，文脈から社名か機器名かなどを判断する必要がある。普通，大文字で始まり Corporation や，Inc. と続く場合は社名と考えて良い。例えば，General Terminal Corporation は，ジェネラルターミナル社である。機器名は全て大文字で書かれているこ

とが多い。例えば，GT-100/GT-400 Display Terminals である。これは，GT-100/GT-400 が機器名，Display Terminal が表示装置を示している。またソフトウェアなどは，フルネームの記述後省略形が記述されることが多い。Distributed Systems Environment (DSE) などである。

[メ モ]

(41) 次世代のコンピュータ

Ministry of International Trade and Industry(MITI) and the domestic computer manufacturers will carry out research and development of software for the next generation of computers, under a five-year project that starts in fiscal 1979. Central to the development work is the operating system, and this will be undertaken by a seven company set-up, consisting of a group of five companies that include Fujitsu, plus NEC-Toshiba Information System and Computer Central Laboratory.

(出典：Jipdec Report summer 1979, JIPDEC)

【重要語句】

- carry out　実行する
- research and development　研究開発
- next generation　次世代
- consisting of～　～から構成される
- include　を含む

【解説】 次世代のコンピュータに関しての通商産業省の政策についての記事である。語句，構文ともに易しいので十分理解できる。この程度の文章を数多く読んでおくと良い。以下内容に関連して簡単に補足する。

1979年現在のコンピュータは，3.5世代として位置づけられ，真空管を素子としていた1世代，トランジスタの2世代，IC（集積回路）の3世代からLSI (Large Scale Integration；大規模集積回路）を素子とする時代となっている。そして次世代（4世代）のコンピュータではVLSI（Very Large Scale Integration；超 LSI）を利用するものが一般化されるものと考えられている。

通商産業省（MITI）では国産コンピュータメーカを結集し，次世代の高性能コンピュータの開発を積極的に支援している。なお4世代のコンピュータの利用形態は，エンドユーザ向きの集中・分散処理（End user oriented integrated distributed processing）が基本として想定されている。

本文は日本の動向を海外へ伝えるための英文であり，日本の企業及び団体の

略称形が使われている。以下に本文で取り上げたもの以外の例も含めて挙げておく。

MITI　Ministry of International Trade and Industry　通商産業省

NEC　Nippon Electric Co., Ltd. 日本電気株式会社

NTT　Nippon Telegraph and Telephone Public Corporation
日本電信電話公社

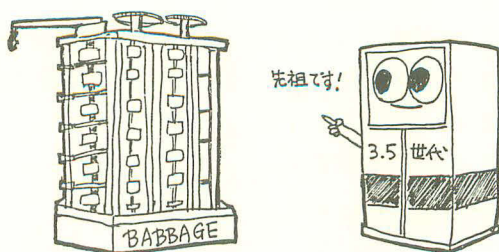
MPT　Ministry of Post and Telecommunications　郵政省

KDD　Kokusai Denshin Denwa Co., Ltd.　国際電信電話株式会社

JEIDA　Japan Electronic Industry Development Association
（社）日本電子工業振興協会

JECC　Japan Electronic Computer Co.　日本電子計算機株式会社

JICST　Japan Information Center of Science and Technology
日本科学技術情報センター

JIPDEC　Japan Information Processing Development Center
（財）日本情報処理開発協会

［メ　モ］

(42) マイクロコンピュータ市場

> The microcomputer market is expanding explosively thanks to the low prices of microcomputers and their broad range of applications. Since their use is not confined to industry alone and they are beginning to play an important role in the home as well, they are coming to have a considerable significance in society in general and have been arousing great popular interest. Japan's microcomputer industry has before it limitless possibilities, although there are also unique problems here as compared with countries overseas.
>
> (出典：Jipdec Report summer 1979, JIPDEC)

【重要語句】

- market　市場
- is expanding explosively　爆発的に拡がる
- applications　適用，アプリケーション
- since　だから
- thanks to　のおかげで
- range　範囲
- confine　限定する
- considerable significance　かなり意義があること
- arouse　喚起する，起こす
- before　目前に
- overseas　海外の

【解説】 我が国のマイクロコンピュータ産業 (Japan's microcomputer industry)について述べたものである。基本的な語句の知識として文脈から since や before の意味を把握しておく必要がある。この場合「以来」,「以前」という意味ではない。

マイクロコンピュータは市民生活の高度化と多様化にともなって，今後は民生分野への市場の拡大が考えられている。ただし，本文にも述べられているように，日本のマイコン産業は海外諸国に比べて独自の問題がある。このことに

関しては，引用した文献の後で述べられているので本文の理解を助けるために参考としてまとめておく。

マイコン産業を構成するのは，**チップメーカ**（chip manufacturer：LSI 素子を作るメーカ），**システムハウス**（system house：ソフトウェアを作る会社），**ユーザ**（user）である。チップメーカの問題点は，マイクロコンピュータはLSI そのものであり，ハードウェアの良し悪しは LSI にかかっているが，大量生産によりコストダウンをはかるためには，限られた企業しか生産できないことである（生産量を倍にすると25% ほど価格ダウンができる）。また，ソフトウェアハウスでは，米国のようにいわゆるベンチャービジネス的な企業がないため，資金やオペレーション費用が増大すると，経営者の判断にもよるが止めてしまう場合がある。ユーザの問題としては，マイクロコンピュータの機能についての知識が不十分，マイコンを組み立てるとき，説明書などを無視して正しい方法と順序で組み立てないなどの問題点がある。いずれにしても，このような問題はささいなことであるから，本文では目前に限りない可能性をもっていると述べている。

［メ　モ］

(43) コンピュータの世代

About every five or six years technological innovation of the basic elements of which computers consist has resulted in the appearance of a computer of an almost completely new type. This time-span of five or six years is referred to as a 'generation'. Developing new systems for the change to a new generation of computers is something that is central to the strategy of the domestic computer manufacturers.

VLSI will be central to the hardware of the new system which will be furnished with an entirely new basic software (OS: Operating System), providing a major increase in cost performance together with a great expansion in utilization capacity.

(出典：Jipdec Report summer 1979, JIPDEC)

【重要語句】

- technological innovation　技術革新
- basic elements　基本素子
- consist of～　～から構成される（～of which computers consist, コンピュータを構成する～）
- result in～　結果として～になる
- appearance of～　～の出現
- refer to～　～を参照する
- generation　世代
- is something that～　～というようなもの
- domestic　国産の（domestic computer manufactures；国産コンピュータメーカ）
- furnish with～　～を備える
- provide　供給する（文中の providing の主語は VLSI である）
- cost performance　費用対効果
- together with～　～とともに

▩ expansion　　拡大

【解説】 コンピュータの世代の概念と，コンピュータを構成する基本素子の開発の動向について述べている。この英文は，実際には GOVERNMENT POLICY（政府の方針）の一部を引用したものである。ここでは，**VLSI** が，基本ソフトウェアを備え，新システムのハードウェアの中核となることを指摘している。

なお，引用文献では，VLSI の性能値が示されているので，参考として以下に示しておく。

VLSI performance

	現在の LSI(Present LSI)	VLSI
容　量 (Memory)	4,000 ビット (4,000 bits)	数百万ビット (Several million bits)
論理回路 (Logic circuits)	100 回路 (100 circuits)	数万回路 (Tens of thousands of circuits)
速　度 (Speed)	8×10^{-6}秒 (8×10^{-6} sec.)	2×10^{-6} 秒 (2×10^{-6} sec.)
大きさ及び重量 (Size and Weight)	ロッカー程度 (Locker-sized)	机上に置けるコンピュータの大きさ (Size of desk-top computer)

表中，大きさ及び重量の項は，LSI, VLSI そのものの大きさではなく，現在の LSI を用いるとロッカー程度の大きさ及び重量をもつコンピュータが VLSI で実現すると，机の上に置けるような大きさになると解釈する。

［メ　モ］

(44) コンピュータシステム一般

Computers and systems applying them have now penetrated into every nook and corner of social life and are playing important roles in the functioning of social mechanisms. Not only are they used in the manufacturing industries, such as in systems of automation for industrial plants; but many computer systems are also being used in areas which have a direct bearing on our own everyday lives. A few of the examples which might be mentioned are seat reservation systems for trains and airplanes, electronic funds transfer systems in banks, and systems for selling train and bus passes.

Our every lives are influenced not only by large-scale systems. Micro-computers have today come to be incorporated in various types of household electric appliance and in automobiles, making it possible to control these products more smoothly and reliably. A wave of computerization is engulfing the life of society as a whole.

(出典 : Jipdec Report winter 1979, JIPDEC)

【重要語句】

- penetrate into　浸透する
- every nook and corner　あらゆるところで
- play important roles in　重要な役割を演ずる
- not only～but also—　～だけではなく—も
- plant　工場
- seat reservation system　座席予約システム
- electronic funds transfer system　電信振込システム

【解説】 基本的なイディオム及び単語の知識があれば，主旨は十分理解できる。every nook and cornerのような慣用的な表現はなるべく多く憶えておくと良い。

コンピュータ及びこれを中核としたアプリケーションシステムは今や社会生活の中にすみずみまで行きわたりつつあり (penetrated into every nook and corner of social life)，運輸・銀行関連のシステムなど我々の生活に欠かせない

場で，これらが重要な役割を果たしている。座席予約システム（seat reservation systems）や銀行の振込システム（electronic funds transfer systems）はその典型的な例である。

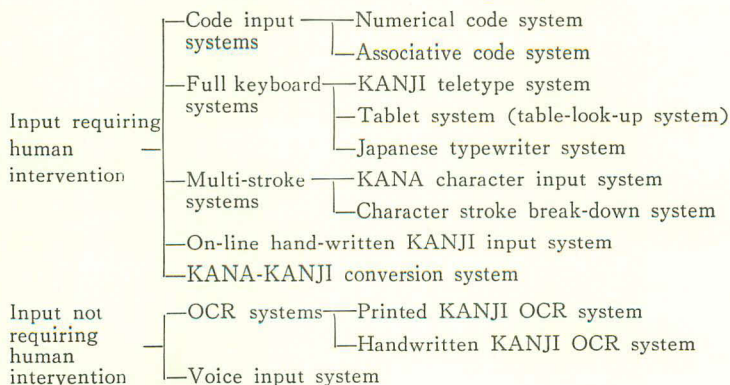
しかし，我々の日常生活がこれらの大型システムによってのみ影響を受けるのではない。マイクロコンピュータの出現によって，家電製品や自動車にまでコンピュータが使われていると説明されている。

［メモ］

(45) 入力システムの動向

In data processing the fundamental method for data input is to press keys on a keyboard. However, in input of Japanese-language information, it must be possible to supply input of thousands of different characters. It is no easy matter to arrange so many characters on a key-board and to select the necessary keys.

For this reason, research and development activities have been under way for the various input systems classified in Table 1. We have not yet reached the point of settling on a standardized system such as the one used for inputs of English alphabet and numerals.

Table 1 Classification of input systems

(出典 : Jipdec Report winter 1979, JIPDEC)

【重要語句】

- fundamental method 基本的方法
- Japanese-language information 日本語情報
- it must be possible to～ ～可能であるに違いない（できなければならない）
- under way for～ ～について進行中
- have not yet reached まだ到達していない

▨ standardized system　　標準化されたシステム

【解説】 論文としては，専門用語が少なく理解し易いが，非常に重要なことを述べている。**日本語情報処理システム**（Japanese-Language Information Processing System）は我が国では潜在ニーズは十分あるものの，日本語の特殊性のゆえ，未だにデータの入力の問題がネックとなっている。英文タイプライタのように本当にユーザにとって使い易いシステムはまだ研究段階であり，種々の方式が検討されているという主旨である。

なお，表は入力システムの分類で，現在よく使われている順番に並べられている。コード入力システムは，漢字に対応するコードを入力する方式である。また，タブレットシステムとは，漢字鍵盤の該当位置をペンのようなものを当てる方式である。それゆえ**ペンタッチ方式**とか，該当漢字を拾うということから**ルックアップ方式**ともいわれる。

［メ モ］

(46) 技術革新

In today's world, great amounts of information circulate through a wide variety of industries and assist in promoting industrial activity. At the same time, there can be seen a remarkable and rapid growth in computer utilization. Computers have, for example, been applied even to home electrical appliances such as air conditioners and television sets to greatly improve their practicality. There has also been seen a rapid dissemination of compact computers as is reflected in the so-called "micro-computer boom."

(出典：Jipdec Report winter 1979, JIPDEC)

【重要語句】

- great amounts of～　非常に大量な～
- information　情報
- circulate through～　～の中をめぐる
- at the same time　同時に
- remarkable and rapid growth　目を見張るような急激な成長
- have been applied even to～　～にさえも応用されてきた
- such as～　～のように
- dissemination　まき散らすこと，流布
- is reflected　反映されている
- so-called　いわゆる

【解説】 本文は，最近のマイクロコンピュータブーム（micro-computer boom）といわれるようになった背景について述べたものである。内容的には，極めて常識的なものであり，【重要語句】に示した程度の単語の知識があれば十分に理解できる。

現在は，情報処理という概念が，社会の中では，コンピュータの利用（computer utilization）という形で具体化されつつある。従来のコンピュータという言葉が大型コンピュータを意味していたのに対して，最近では，例えばマイクロコンピュータの出現により，家電にも応用され(home electrical appliance),

情報処理という概念が具体的になってきたことを示している。すなわち，日常生活にコンピュータが現実に侵透してきたといえる。

本文では述べられていないが，このようにコンピュータが普及した原因は，ハードウェア技術の進展により，LSI (Large Scale Integration；**大規模集積回路**）など，コンピュータ素子（basic element）の超小型化，低価格化がはかられたことである。さらに，personal computer（**パーソナルコンピュータ**あるいは**パソコン**と呼ばれる）なるものが出現し，ユーザが自由に組み立て，処理を行うことのできるようなものも出現している。

〔メ モ〕

(47) 印刷装置の動向

Line printers, serial printers, and CRT displays have been used in the past as output devices for the results of information processing in the form of combinations of English alphabet, numerals, and KANA. These output devices are also required for the output of Japanese-language information, and large numbers of devices having a wide variety of different functions have been developed by manufacturers.

The following factors may be mentioned in connection with KANJI printers: character quality, number of characters, printing system, printing speed, and paper used for output. Since the number of the characters printed may amount to 10 to 100 times the number of those used in printing the English alphabet, numerals, and KANA (128 characters), the character generator will have an immense size. The dot composition and density needed to represent the individual will also contribute to this large size, and the printing speed will decrease as a result.

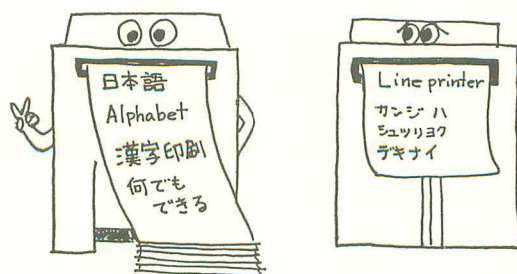
(出典：Jipdec Report winter 1979, JIPDEC)

【重要語句】

- line printer　ライソプリンタ，行印字機（1行ずつ印刷する装置）
- serial printer　シリアルプリンタ（1文字ずつ印刷する装置）
- CRT display　CRT (Cathode Ray Tube：陰極線管)ディスプレイ
- in the past　過去に
- output device　出力装置
- function　機能
- combination　組合せ
- large numbers of　非常に多くの
- manufacturer　製造者，メーカ
- since　〜なので

▩ on account of　　～のために，～という理由で

【解説】 日本語情報処理における出力装置（output devices）に関する論文の一部である。内容的に深くふれていないため語句の基礎知識があれば十分理解できる。

入力装置と同じように出力装置もエンドユーザとのインタフェースなどを考慮し，最近ではメーカがすぐれたものを発表してきている。しかし日本語情報処理においては，その特殊性からさらに検討すべき課題が存在していると述べている。なお，本文で dot とあるのは，漢字を点の集合とみなして印字する方法で，従来の活字では文字種が限られるが，**dot 方式**では，広範囲の文字を印刷することができる。

[メ　モ]

(48) 第4世代のコンピュータ

The 4 th generation system will require major functional advances in software. Japan is said to be lagging even further behind in software than it is in hardware, and this makes it all the more imperative that it develops basic software technology (operating system) such as for network administration and ultra-high-level language processing.

Furthermore, with network being the main type of processing mode with the 4 th generation, the role of peripheral and terminal equipment will be very important, meaning that at the same time new technologies will be needed to produce such devices.

(出典 : Computer White Paper 1979 Edition, JIPDEC)

【重要語句】

- lag　遅れる
- behind in～　～に遅れている
- imperative　緊急な
- ultra-high-level language　超高級言語
- role　役割
- peripheral　周辺の
- equipment　装置

【解説】 次世代のコンピュータに関しての技術的な要求に関する説明文である。英単語に関する知識だけでは，主旨を完全に理解することはむずかしい。技術動向については常に把握しておく基本姿勢が必要であろう。第4世代のコンピュータの利用形態は高度な**分散処理**（distributed processing）が第1に要求されており，このために相対的にソフトウェアの高度化が重要となってくる。すなわち，データベースの分散技術，エンドユーザ向きの**超高級言語**（ultra-high-level language），ネットワーク管理（network administration），日本語処理技術の高度化が必要である。またエンドユーザに向いた処理を前提とするため，**周辺端末装置**（peripheral and terminal equipment）の役割はデータ

の入出力という点で重要となり，多種多様な使いやすい機器の開発が望まれている。

なお，本文で出てくるコンピュータの世代は，主として，使用される素子の観点からとらえられたものであり，一般的には，次のように分類されている。

第1世代　1950年頃から1960年頃までの期間で，真空管回路が用いられ，記憶装置としてブラウン管記憶装置，遅延線，磁気ドラム，入出力装置として紙テープやカードが用いられた。

第2世代　1960年頃から1964年頃までの期間で，トランジスタ回路が用いられ，記憶装置として磁心記憶，入出力装置として磁気テープが主流であった。

第3世代　1965年頃から1970年頃までの期間で，ICが用いられ，磁気ディスクを中心とした補助記憶装置が用いられていた。

第3.5世代　1970年頃から現在に至るまでの期間でLSIが用いられている。

第4世代　本文の主旨でもあるが，1980年代が相当するといわれている。この世代では超LSI（Very Large Scale Integration）が中心となっている。

［メ モ］

第Ⅱ部　応用編

　情報処理技術の水準をはかるバロメータの１つとして例年通商産業省が実施している情報処理技術者試験が挙げられる。この試験の午前の問題に各種別（第２種，第１種，特種）ごとに，２題ずつ英語の問題が出題されている。

　応用編では，過去に出題された問題のうち比較的平易な英文が用いられている第２種を中心に英文を選択し，ソフトウェア，ハードウェア及びその他に分類し，それらの問題の解説，解答を示した。解説においては，問題の解法を述べた上で，関連する知識もあわせて説明した。

　応用編の主な目的は，英文の読解力とコンピュータの基礎知識及び関連する知識を実践的に得ることである。

第5章　ハードウェア

(49)　計算機用語―1―

次の英文は，電子計算機の用語に関する説明の一部分である。それぞれの文に最も関連のある用語を解答群の中から選べ。

a　The wide usage of them in third-generation computers reduced the circuit design tasks to those of generating electrical compatibility between circuits and calculating their delays, noise margins, junction temperatures, and the like.

b　Data is stored in tracks that are concentric*1 circles on the surface. It is divided into sectors, i.e. pie-shaped sections. Hence the tracks are divided into arcs, each arc being the intersection of a track with a sector.

c　It can perform the arithmetic and logical operations on numeric and nonnumeric data. It has one or more accumulators to store the data in advance to the operation.

d　There are several types of transports. Each type has two versions, for recording information in nine tracks and seven tracks. Each transport can record information in both low and high densities, 556 and 800 bits per inch.

(注　*1＝同一中心の)

解答群

ア　core memory　　イ　buffering
ウ　disc storage　　エ　integrated circuit (IC)
オ　transistor　　カ　cathode ray tube (CRT)
キ　magnetic tape　　ク　graphic display
ケ　central processing unit (CPU)

(第2種　昭和51年)

【解説】 以下に解答群の用語について説明する。

core memory　magnetic core memory ともいう。コンピュータの主記憶装置 (main memory, main storage) を構成する。

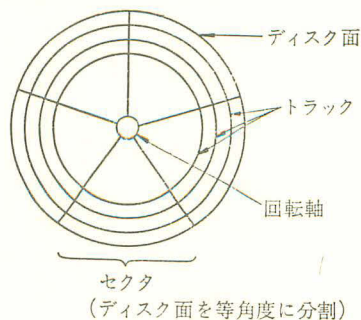
buffering　緩衝手法あるいは緩衝方式と訳されるが，一般的にはバッファリ

ングと呼ばれる。動作速度の著しく違う装置間（例えば，入出力装置と主記憶装置）でのデータ転送の調節を行う技法。

disc storage 補助記憶装置のディスク面。

セクタとトラックの関係は，概念的には下図の通りである。

integrated circuit（**IC**） 集積回路と訳し，一般的には IC と呼ばれている。これはトランジスタと違い，同一規格のものが大量生産できる。

transistor 半導体を利用した回路素子。真空管に比べて利点は多いが，使用周波数が少ない，温度による特性の変化が大きいなどの欠点がある。

cathode ray tube（**CRT**） 本来は，陰極線管のことであるが，これを使用した文字表示装置に対して，CRT ディスプレイという表現で用いる。

magnetic tape 磁気テープ装置のこと。問題では，556 BPI, 800 BPI となっているが現在では 1600 BPI が普通である。BPI は bit per inch の省略形。

central processing unit（**CPU**） 中央処理装置と訳し，CPU と呼ぶ。

【解答】 a. エ　b. ウ　c. ケ　d. キ

［メ モ］

(50) 中央処理装置 (CPU)

次の英文中の［　　］に入れるべき適当な字句をそれぞれに対する解答群の中から選べ。

a　The ［ a ］ section is responsible for decoding computer commands.

b　Index registers are ［ b ］ under the control of the programmer.

c　The address register in the control section is used to hold the memory address of the ［ c ］

d　During the ［ d ］ cycle, the control section is responsible for getting the next instruction from computer storage.

e　Index registers are used to store memory address to modify the ［ e ］ portion of a computer instruction.

a，b，c に関する解答群

ア　input output　　イ　never　　ウ　last instruction executed

エ　control　　オ　sometimes　　カ　next instruction to be executed

d，e に関する解答群

ア　fetch　　イ　operand　　ウ　command　　エ　execute

(第2種　昭和52年)

【解説】 a．コンピュータコマンド（コンピュータに対する指示）を解読 (decode) するのは，中央処理装置の制御装置で行う。この英文では装置という表現ではなく，部 (section) が用いられている。従って，ここは control section が適当である。

b．index register は，指標レジスタといわれることもある。index register は，アキュムレータなど他のレジスタと同様に演算操作を行うことも可能であるが，最大の特徴は，命令の指定により，この内容が命令のアドレス部に加えられ，その結果が実際のアドレス（実効アドレス： effective address）として用いられることである。解答群の中から適当な語をさがすと，sometimes が該当する。他に適当な語はない。sometimes にすると，index register は時にはプログラマが，使えるという表現になる。

c．アドレスレジスタは，中央処理装置の制御部にあり，次に実行すべき命令の記憶装置（main memory）での番地（address）を記憶しているレジスタである。また実行するために，記憶装置から読み出された命令を受け取って一時的に記憶するレジスタを命令レジスタ（instruction register または control register）という。命令の解読はデコーダ（decoder）が行う。

d．あるサイクルにおいて，記憶装置から次の命令を取り出すとある。このサイクル（一定時間）のことを fetch cycle という。正確には，fetch は，主記憶装置またはレジスタにデータやプログラムルーチンを読み込むことをいい，fetch cycle とは，コンピュータの制御装置が前の命令の実行を終了してから，次の実行すべき命令を記憶装置から取出し終えるまでの動作段階のことをいう。fetch cycle は，命令取出し段階と訳す。fetch cycle に対して execution cycle がある。これは，コンピュータが新しい命令を受け取ってから，その命令の実行が終了するまでの動作段階のことをいう。execution cycle は，命令実行段階と訳す。

e．インデックスレジスタによる修飾について述べたものであり，bで説明したように，アドレス部の値を変更する。一般的に，アドレス部は，命令（instruction）のオペランド部ともいう。

【解答】 a. エ　b. オ　c. カ　d. ア　e. イ

［メ モ］

(51)　命令 (INSTRUCTION)

次の英文中の[　　]に入れるべき適当な単語を解答群の中から選べ。

The ability of computers to execute conditional-branch instructions —i.e., to [a] the flow of control of their programs as a function of the [b] of tests on intermediate results of their own computations —is one of their most [c] properties, for every effective procedure can be reduced to a series of nothing but commands (i.e., statements of the form "do this" and "do that") interlaced with conditional-branch instructions. Moreover, only [d] branching instructions (i.e., instructions of the form "if such-and-such is true, do this; otherwise do that") are needed. If the decision whether such-and-such is true or not itself involves a complex [e], this too can be cast into the framework of commands and two-way branching instructions.

解答群

ア	reason	イ	binary	ウ	modify	エ	irrelevant
オ	declaration	カ	process	キ	outcome	ク	monitor
ケ	crucial	コ	symbolic				

(第1種　昭和52年)

【重要語句】

- moreover　　その上
- such-and-such　　しかじかの，何々の
- framework　　体系

【解説】 分岐命令及びそれに関連する手続きについて述べた文である。コンピュータの全ての処理がコマンドと2値分岐命令にすることができると述べている。条件付分岐命令とは，本文の説明にもあるように，設定した条件が成立したとき（真）だけ分岐を生ずる命令である。条件が成立しないとき（偽）は，この命令は無効になり，そのまま続行されるのでプログラムの流れが条件によって2つに分岐する。なお，branch は，そのまま，ブランチと呼ばれることも多い。

以下に，空白を含む文節ごとに検討する。

a．b．「条件付分岐する命令を実行するためのコンピュータの能力」をいいかえている部分である。分岐命令は，条件によってプログラムの流れを変える命令であり，条件はプログラムの処理の過程で発生する。従って，aは変更するという意味の modify，b は調べた結果の機能という意味から outcome が該当する。

c．「最も□な特性である」であり，次の for 以下で分岐命令の重要性について述べていることから crucial（重要な）が該当する。

d．すぐあとのかっこの中の説明が，「□ branching instruction」をいいかえている。かっこ内は，「もし真ならばこれをせよ，そうでなければあれをせよという形式の命令」という意味である。これは，条件を設定し，この条件が満たされた場合と満たされなかった場合の2つの状態を設定し，各々処理を行う命令である。従って binary（2つの）が該当する。

e．「判断の結果はどうであれ，それ自身が複雑な□を含んでいれば，これもまたコマンドと2つの分岐する命令の体系に組込むことができる」となる。この文節の前で，分岐命令は，binary branching instruction だけで良いと述べられ，この文では，複雑な□もまた，command と two-way branching instruction となると述べられている。従って，複雑な手続き（処理）という意味である。なお，この場合 two-way は binary と同じ意味である。

【解答】 a. ウ　b. キ　c. ケ　d. イ　e. カ

〔メ モ〕

(52)　中央処理装置（CPU）と印刷装置

次の英文の内容に最も近い文章を解答群の中から二つ選べ。

If CPU design is the hare of the computer industry, printer design must be the tortoise; the former races forward almost yearly, while the latter makes steady though hardly dramatic progress. Although the speed of printers has picked up somewhat, the limiting factor is still mechanical technology. Printers are slow, produce heat and are subject to problems such as friction and wear. But all this is expected to change by 1985, when as much as 70% of the printers shipped will be based on other technologies, notably ink-jet and laser-xerographic technology. The driving force behind this change will not be the fast-growing field of word processing, but rather the rise of electronic mail and its associated networks. It can be predicted that the most important trend in the 1980's that will affect printers will be the increased intelligence of networks, terminals and printers coupled with massive, easy-to-use data base and electronic file systems.

解答群

ア　印刷装置の技術は，カメがウサギを追い越したように，今や中央処理装置の技術に追いつき追い越してしまった。

イ　印刷装置には機械的可動部分がつきもので，これが印刷速度の向上を妨げている。

ウ　近年急激に進歩している文章処理技術に促され，1985年ごろまでには，大部分の印刷装置はインク噴射方式やレーザ印刷方式を採用するようになると予想される。

エ　将来の印刷装置のあり方は，通信網やデータベースシステムと切り離して考えることはできない。　　（第1種　昭和54年）

【重要語句】

- while　ところが一方
- though　といっても
- pick up　だんだん良くなる
- somewhat　多少
- friction　摩擦
- wear　摩耗
- as much as　～ほど
- xerographic　電子写真の

▨ predict　予言する　　▨ trend　動向

【解説】 印刷装置の技術動向に関して述べた英文である。以下，本文で使用されている技術用語の説明をした後で，誤っているア，ウの理由を示す。

mechanical technology 機械技術ということであるが，具体的には，歯車など機械可動部分のことをいう。印刷装置でいえば，活字をたたいて印刷するような機構である。従って，当然，摩擦，摩耗などが起こる。

ink-jet ノズルから噴出したインク粒子に電荷を与え，これを電界により偏向してドット方式で印刷する方法。mechanical ではないので耐久性がある。

laser-xerography（本文では laser-xerographic となっている） 電子写真法（光波を電磁波に変え，インキ粉を散布した複写紙に当て熱処理によって感光定着させる方法）の一種で光波としてレーザを利用する。

ア. 「毎年レースで前進しているのは CPU」という記述がある。

ウ. 「文章処理技術（ワードプロセッシング）の分野の急成長ではなく，むしろ電子郵便（electronic mail）と，それに関連する通信網（ネットワーク）……」という記述に反する。

【解答】 イ　エ

［メ モ］

(53) タイムシェアリングシステム

次の英文中の[ ]に入れるべき適切な単語を解答群の中から選べ。

Time sharing provides a mechanism to split computations into slices and allocate only the critical resources to the [a] computations according to rules which attempt to strike a balance between the productivity of users and the system. The user at the [b] can now be considered also as a critical resource. Unfortunately, users are very slow in execution and demanding when they require computation. Fast response for such users is provided by limiting the competing computational slices to a fraction of a second by means of a timer-driven [c] When users signal that they are ready to use the computer, their process will be put in the [d] and executed after a very short delay. The hardware resources which are most critical in most time sharing systems are the CPU and main memory. During [e] periods the users' process may be swapped out of the primary resources.

解答群

ア	interrupt	イ	paging	ウ	peak	エ	queue	オ	active
カ	center	キ	memory	ク	terminal	ケ	complex	コ	inactive

(第1種 昭和54年)

【解説】 以下，用語について説明したあと，空欄について検討する。

time sharing system（TSS） 多くの端末装置を通信回路などで中央側の計算機に接続し，多くの利用者が同時に計算機を利用する方式。time slice（時分割と訳し，ある仕事に対して一定時間処理すると次の仕事の処理を行い，さらに，別の仕事を一定時間処理を行うというように，複数の仕事に対して極めて短い時間を与え，結果として同時に複数の仕事が行われているようにみえる）及び会話型言語が採用されている。

interrupt 割込みと訳すが，インタラプトということが多い。実行中のプログラムを再実行可能な状態にし，実行を停止させること。割込みの原因としては入出力装置に対する指令の終了，CPU 内部での誤動作などがある。

queue　待ち行列と訳し，キューあるいは単に待ちということが多い。サービスを受けようとして待っている人の集まりのことから転じて処理の開始を待っている仕事の集まりのことをいう。ジョブキュー(job queue)ともいう。

swapped out　優先度の高いプログラムを主記憶装置上に配置するため，優先度の低いプログラムを補助記憶装置などへ移すこと。この英文では，ユーザが考えている間，処理が行われていないわけであるから，補助記憶装置に移されるという意味で使っている。

a．ユーザが行う演算の形容である。計算が複雑（complex）かどうかは主観によるものであるから活発（active）の方が適切である。

b．ユーザのいる場所である。TSS は端末から行うことを考えると，端末（terminal）が該当する。

c．time slicing のことであり，一定時間単位に処理を行うのであるから，中断させる機能のことである。TSS では timer により管理（タイマーインタラプト：timer interruption：計時機構割込み）する。

d．実行まで少し待たされるとある。従って，queue が適当である。

e．ユーザが考えている状態（CPU など計算機の資源は使用していない）であるから inactive が適当である。

【解答】 a. オ　b. ク　c. ア　d. エ　e. コ

［メ モ］

第6章　ソフトウェア

(54)　プリプロセッサ

次の英文の意味に最も近いものを解答群の中から一つ選べ。

SIMPLE is an independent COBOL preprocessor producing COBOL source program. A modular file matching package is generated for sequence checking of as many as six files. A sequential or indexed sequential processing logic can be generated. A control break detection of up to ten control levels with branching to the appropriate user-defined routine as the control breaks are sensed is featured.

解答群

ア　SIMPLE は COBOL から呼び出すサブルーチン群で，順ファイルと索引順次ファイルを処理することができる。

イ　SIMPLE はユーザの指定に従って，ファイルの突き合せ等を行う COBOL ソースプログラムをつくりだす。

ウ　SIMPLE は最大6個のファイルの突き合せを行うパッケージで，COBOL ソースプログラムの形で提供される。

エ　SIMPLE の実行中コントロールの切れ目が10回以上起こると，あらかじめ指定したユーザのルーチンに移る。　(第2種　昭和51年)

【重要語句】

▨ up to～　　～まで

▨ feature　　特徴づける

【解説】　以下に解答群の内容を順に説明する。

ア.「SIMPLE は，COBOL ソースプログラムをジェネレート（作り出す）するための」とあるからこの記述は誤りである。「COBOL から呼び出すサブルーチン群」は，一般的には library と呼ばれる。この文の後半の「順次ファイルと索引順次ファイルを…」の部分は，全訳では「順次処理あるいは索引順次処理」と訳しておいたが，ファイルのシーケンスチェック（順序検査）を行うプログラムがジェネレートされるわけであるから，ファイル処理と考えて良い。なお，ジェネレート及びシーケンスチェックは日本語に訳すよりも，

このまま英語で表現される。また，プリプロセッサとは，前処理プログラムと訳され，ある特殊な機能を付加するため，本処理の前に行う処理をするためのプログラムである。これも，プリプロセッサと呼ばれることが多い。

イ．この通りの記述がある。

ウ．随所に generate という表現が出てくるが，これは，「作り上げる」という意味である。一般的には，ある条件を与えることによって，自分の希望するプログラムを作り上げることを generate，また，作り上げるプログラムのことを generator という。全訳で分るように，「SIMPLEは，最大6個のファイルの突き合せを行うプログラムをジェネレートする」とあり，パッケージそのものではない。従って，この記述は誤りである。ジェネレートされた結果が COBOL ソースプログラムであることに注意する。

エ．本文では，「制御の切れ目が検出されると，ユーザが定義したルーチンへ分岐し……制御の切れ目は最大 10 レベル」とある。すなわち，制御の切れ目が 10 回起こったらという条件ではない。従って，この記述は誤りである。制御の切れ目（control break）は，一般に，英語そのままで，コントロールブレークと呼ばれる。control break は，「control break を起こす」という表現で用いられることが多い。これは，例えば，キーとなる項目の内容の昇順にデータが並べられているような場合，最初からその内容をチェックして行って，値が変化したときの状態をいう。また，ここでいうレベルとは，キーとなる項目の数のことである。

【解答】 イ

〔メ モ〕

(55) 言語一般

次の英文中の[   ]に入れるべき適当な単語を解答群の中から選べ。

A typical classified section of the *Sunday Los Angeles Times* contains perhaps 50 ads for programmer jobs that mention [a], 5 that mention [b], and 1 that mentions PL/I, with seldom any mention of ALGOL, BASIC, or APL. This is especially significant because Los Angeles is the heart of the aerospace industry where scientific languages such FORTRAN, PL/I, and APL should be used the most. Certainly COBOL is not so widely used because of indoctrination; universities usually teach [b], frequently PL/I, ALGOL, BASIC, and APL, but rarely COBOL. [a] is widely used because it is better suited to the type of applications to which the computing industry has moved, and most graduates must learn [a] outside of the universities in order to get jobs in programming.

解答群

ア	APL	イ	ASSEMBLER	ウ	COBOL
エ	FORTRAN	オ	MACHINE LANGUAGE		
カ	BASIC	キ	RPG		

(第2種　昭和54年)

【重要語句】

- aerospace　航空宇宙
- because of　～なので
- indoctrination　教えること

【解説】 COBOL と FORTRAN の使われ方について論述している。航空宇宙産業の中心地であるロサンゼルスでは，本来科学技術計算言語に関する求人が多いと思われるのにその反対である。一般的に COBOL は，事務処理に適した言語として開発されたものであり，FORTRAN は，科学技術計算用として開発された言語とされているので，aが COBOL，bが FORTRAN となる。さらに，内容を確認してみよう。次の段落で，大学では，COBOL はほとんど教えない，FORTRAN か，PL/I，ALGOL，BASIC，APL など

を教える，という記述も現実によくあることである。さらに，一般の計算機業務（computing）に適合する言語という部分もCOBOLのことである。科学技術計算はやはり，特殊な分野と考えて良いであろう。だから，COBOLの求人の方が多いのだと論じている。

以下に，FORTRANとCOBOL以外の選択肢について簡単に説明する。

APL（エーピーエル）　A program language の略で，アイバーソン言語を対話方式にしたもの。

ASSEMBLER（アセンブラ）　比較的機械語に近い構造をもつプログラム言語で，文法は機械語に近いが，命令コードや演算子などは記号で表現する。一般には，アセンブラの1命令は1機械語に対応する。

machine language（マシンランゲージ）　機械語と訳す。これは，コンピュータが直接解読できる言語で，2進数を組み合せて記号化されている。

BASIC（ベーシック）　Beginner's All Purpose Symbolic Instruction Code の略。ダートマス大学が開発したFORTRANに似た会話型言語で，パーソナルコンピュータに広く用いられている。

RPG（アールピージー）　report program generator の略。報告書などの作成において，必要なプログラムの作成を容易にする。

【解答】　a. ウ　　b. エ

〔メ　モ〕

(56)　プログラミング

次の英文中の□に入れるべき適当な字句を解答群の中から選べ。

Good problem definitions are vital to the [a] of good problems. An incomplete definition implies that the complete structure of the problem is not [b] understood. Missing information, ignorance of [c] cases, and an abundance of extraneous* information in a definition are good signs of [d] programs, or at the best, of programs that will be a [e] to end users.

*extraneous：無関係な

解答群

ア	fully	イ	error	ウ	poor	エ	excellent
オ	surprise	カ	construction	キ	special	ク	service

（第2種　昭和55年）

【重要語句】

- definition　定義
- vital　極めて重大な
- poor　不十分
- at the best　良くみても
- surprise　意外なこと
- excellent　すぐれた

【解説】 プログラムを作成する場合の基本的な姿勢を論述している文章である。解答群をみると分るように，計算機に直接関連するような単語がないので，空欄は，前後関係を理解した上で，適当な単語を挿入する必要がある。以下，空欄単位に適当な語を検討する。

a，b．　a，bを含む文を直訳すると，「良い問題の定義は，良い問題の[a]に対して重大である。不完全な定義は，問題の完全な構造が[b]理解されないことを意味する」となる。bに適当な語は，文法的にみると副詞（adverb）であり，解答群では，fully（十分に）以外は該当しない。また，意味的にも fully で通じる。すなわち，「…が十分に理解されない…」となる。また，一般的に考えても，定義が不完全であれば，意志が十分伝わらないこ

とは明らかであろう。一方，aに適当な語は名詞(noun)であり，解答群では，error(誤り)，surprise(意外なこと)，construction(構築)，service(サービス)などがあるが，aを含む文をbを含む文でいいかえていること，bを含む文は逆にいうと，良い問題の定義が，問題の構造を明確にするという意味になるから，aは「作成」という意味の語が適当である。すなわち，constructionが適当である。

c．cを含む文では，「情報の不足」，「[c]場合の無知」，「無関係な大量の情報」と，良い問題とはならない要素を並列的に並べている。文法的には形容詞(adjective)であり，poor，excellent(すぐれた)，special(特別な)があてはまるが，special case(特別な場合)とするのが最も適当であろう。他の単語では，どのような場合かはっきりしない。

d．dを含む文を直訳すると，「…は，[d]プログラムのgood signsである」という意味になる。……の部分はcを含む文の悪い例の列挙である。ここでは，形容詞でかつ否定的な意味をもつ単語でないとおかしい。従って，poor(不十分な)が該当する。

e．eを含む文を直訳すると，「…あるいは，良くみても，エンドユーザに対して[e]であるプログラムのgood signsである」となる。dを多少やわらかくした表現であること，エンドユーザからみた場合のことを指しているわけであるから，第三者が「予想もしないこと」という内容となる。従って，surpriseが該当する。

【解答】 a. カ　b. ア　c. キ　d. ウ　e. オ

［メ モ］

(57)　在庫管理モデル

次の英文の内容に最も近い文章を解答群の中から二つ選べ。

Historically, a large proportion of data processing activity has been centered on structured problem solving, such as account payable, billing, and payroll. Fifteen years ago, inventory control was considered a highly complex, unstructured problem requiring a great deal of individual judgement. It was generally felt that each organization had a unique inventory problem that could be adequately dealt with only after years of experience at that company. The development of mathematical models that were applicable to inventory problems radically changed the situation. With a successful application of these models, inventory control is now considered to be operational planning of a highly structured nature. On the other hand, there are still large area of organizational decision making, especially in long-range or strategic planning, that are unstructured.

解答群

ア　在庫管理の数学的モデルは，在庫高の急激な変化に対しても有効である。

イ　かつて在庫管理は，各企業が永年の経験にもとづく独自のやり方で処理していた。

ウ　以前には構造が明確でなかった問題でも，今日では構造が明確になっているものもある。

エ　料金計算，給与計算，在庫管理などに見られるように，今日のデータ処理ではあらゆる問題の構造が明らかにされている。

オ　長期的な戦略計画などの組織的意思決定の分野では，今でも数学的モデルが利用され成功をおさめている。　(第1種　昭和55年)

【重要語句】

- account payable　支払業務
- billing　伝票発行（業務）
- payroll　給与計算
- years　（複数形のとき）多年
- on the other hand　一方
- strategic　戦略的な，(戦略上)重要な

【解説】　コンピュータ化の対象業務となり得るには，現状の処理手順を標準

化しやすいこと，すなわち，処理対象となる事象が構造化されていることが必要である。このような観点からいえば，経営における戦略的計画（strategic planning）などにおける意思決定を行う分野は，いまだにコンピュータ化が困難であるという一般的事実を述べた英文である。

strategic planning に関しては，我が国では計算機メーカあるいは大学などで，OR の１つの分野として研究され，経営計画システム，経営意思決定システムとしてさまざまのモデルが発表されており，現在は，種々の数学的手法を用いて過去の実績から将来を予測した結果をグラフなどに表現できるようになっているが，最終決定は人間の判断にまかせられている。在庫制御に関しては定量発注法と定期発注法という方式により，従来の各品目管理から，全ての品目に共通な考え方を適用している（問題が体系化されてきたことを意味する）。

以下に解答群の各選択のうち，不適当なものについて理由を示す。

ア「在庫高の急激な変化に対しても有効である」という記述はどこにもない。

エ「戦略計画のように体系化されていないものもある」という記述が最後にある。すなわち，全ての問題の構造が明確にはされていない。

オ　最後の文節で，戦略計画は，構造化されていないと述べられている。

【解答】 イ　ウ

［メ モ］

(58) プログラムの変更

自分自身を変更するプログラムに関する次の英文の内容に最も近い文章を解答群の中から二つ選べ。

Almost every programming language allows the programmer to modify his program as it executes. In an assembly language program, this practice has an obvious form; in the high-level languages, program modification is sometimes obscured by such fascinating phrases as "program switches," and so forth. One of the disadvantages of programs that modify themselves is their usual inability to be executed more than once. If we wish to execute a self-modifying program a second time, we must load a fresh copy of the program into the machine. This is not necessary, of course, if the programmer takes the trouble to make his program self-initializing. Note that we can have the same problem with variables: If the programmer uses the DATA statement in FORTRAN, or the VALUE clause in COBOL, then a fresh copy of the program must be loaded into the machine for each execution. This is generally not true of PL/I; variables declared with the INITIAL attribute are initialized upon entry to the block in which they are declared, but this adds extra overhead to the program.

解答群

ア プログラムは，そのプログラム中で命令を変更していなくても，一度実行すると最初の状態とは異なるものになることがある。

イ アセンブラ言語を使えばプログラムの実行中にプログラムを変更することができるので，この機能を上手に利用すべきである。

ウ COBOL, FORTRAN, PL/I などの高水準言語ではプログラムスイッチが使える程度なので，実行時にプログラムを変更することはできない。

エ 実行中に内容が変化しても再使用可能なようにプログラムを作ることができるが，そのようなプログラムを同時並行に実行することはできない。

オ 初期値属性を持つ PL/I の変数は，その変数が定義されたブロッ

クに入るごとに初期値が設定される。　(第1種　昭和53年)

【重要語句】

▨ and so forth　など　▨ more than once　2度以上

▨ of course　もちろん　▨ variable　変数

【解説】 以下，選択肢ごとに検討する。

ア　COBOL, FORTRANでは，変数に初期値を与えても，値が変更されることがあるが，PL/I ではそのようなことがないという説明がある。

イ　機能があることの説明はあるが利用方法までは言及されていない。

ウ　逆のことが記述されている。

エ　これに関する記述はない。

オ　これと同じ記述が15行目〜16行目にある。

【解答】 ア　オ

〔メ モ〕

(59) システムソフトウェア

システムソフトウェアに関する次の英文を読み，その内容に最も近い文章を解答群の中から二つ選べ。

The system software is a fundamental part of the total computer system, providing an interface between the user needs and the capabilities of the hardware. A job that is normally done on an elaborate system can be done on a very simple computer with few or no system programs, provided only that there is sufficient memory space and time available. However, the amount of user time involved in preparing such programs as well as the difficulty of checking them can make the task prohibitively expensive. It is often pointed out that the computer can do nothing that cannot be done by hand; that it cannot, therfore, provide the human with powers that he did not previously have. It has also been pointed out that the car does not enable the human to go anywhere he could not have previously walked to, but it *does* enable him to go to many more places a lot more quickly. The car is only about 10 times faster than walking; the computer is about 10^7 times faster than hand calculation. The development of automatic programming systems over the last 15 years has increased the speed with which programs can be written by a factor of 10 to 10^2. The addition of a good software system to a hardware system enables the user not only to complete some tasks more quickly and more cheaply, but makes the computer solution of other tasks possible.

解答群

ア　人間ができないことは電子計算機でもできないが，システムソフトウェアを備えると，人間の及びもつかないようなことができるようになる。

イ　記憶容量が大きくて，処理時間が速い電子計算機でも，システムソフトウェアをもたないと高度な仕事を処理することはできない。

ウ　過去15年間にわたる自動プログラミングの発展によって，プログラム作成は10～100倍も速くなった。

エ　システムソフトウェアは電子計算機システムの基本的構成要素であり，これによりプログラムの実行や検査が容易になり，かつハードウェアの能力を十分に活用できるようになる。

オ　電子計算機によいシステムソフトウェアが備われば，一部の仕事をより速く，より安価に処理できるようになる一方，仕事によっては処理が可能になるものも出てくる。　　（第1種　昭和51年）

【解説】 以下，不適当な選択肢に関して理由を示す。

ア　及びもつかないことができるという記述はない。

イ　一般的事実ではあるが，この英文では記述されていない。

エ　ハードウェアの能力を十分に活用できるようになるという記述はない。

【解答】 ウ　オ

［メ モ］

(60) ファイル編成

ファイル編成に関する次の説明文を読み，解答群の中からシーケンシャルオーガニゼーションに関するものを2つ選べ。

説明文

sequential organization : Records are stored in positions relative to other records according to a specified sequence.

random organization : Records are stored and retrieved on the basis of a predictable relationship between the key of the record and the direct address of the location where the record is stored.

解答群

ア Any record can be retrieved by a single access.

イ This organization allows rapid access of the next record in the file but offers difficulties in updating a file and retrieving records out of sequence.

ウ The advantage of being able to rapidly access the next record becomes a disadvantage when a file is to be searched until a record having a particular key value is encountered.

エ Individual records can be stored, retrieved and updated without affecting other records on the storage media. (第1種 昭和47年)

【重要語句】

- on the basis of～　～を基礎として
- out of　範囲外
- advantage　有利な
- disadvantage　不利な
- storage media　storage medium の複数形，記憶媒体

【解説】 ファイルの編成は，問題文以外に索引順次編成 (indexd sequential organization) がある。索引順次編成は，データレコードの位置及びデータレコード内のキーの値をもったインデックスレコードが通常のデータレコード以外にあり，このインデックスレコードにより，順次編成，乱編成両方の処理を行えるようにしたものである。

以下に，順次編成，乱編成に関して補足する。

順次編成　常にファイルの先頭からアクセスしなければならない。ただし，読込み，書込みの際，ブロッキングによって比較的高速度処理が行えるので，次のレコードを処理するのは速いが，例えば，10レコード目と500レコード目だけを処理したいような場合は，乱編成の方が良い。なお，アクセス（access）とは，記憶装置に関していうと，読込み，書込みなど何らかの動作を与えることをいう。

乱編成　レコードのキー（レコード中の項目であることが多い）を数学的に変換して番地（記憶装置上での格納位置で直接アドレスのこと）とするもので，キーの変換式を知っていれば，キーの値が与えられたとき，変換式より計算して，該当のレコードの位置を知り，レコードを読み込んだり書き込んだりできる。変換式によっては異なったキーが同一番地に割当てられることもあるので注意が必要である。

【解答】　イ　ウ

［メ　モ］

(61) システム設計

次の英文の内容に最も近い文章を解答群の中から二つ選べ。

One of the functions of the systems analyst is to obtain a requirement from a user and produce an outline design for the system. This design is described in a document often called a user specifications. The user is then asked to approve this document. This approval often represents the last moment that the user can change his requirements without the risk of an expensive redesign. Consequently it is imperative that any omissions or misunderstandings in it are identified and corrected. The user specification must, therefore, be expressed as clearly as possible in terms the user can easily understand. With a small set of batch programs, the user is dealing with a relatively straightforward document describing a small range of functions. However, in a complex system which must handle several thousand data elements and many hundred relations among them, it is improbable that user management will be able to fully understand the details of the proposed design. The details of the design are obtained from the user and, although the designer can ensure that the information is self consistent, only the user can ultimately check the design.

解答群

ア システムの概要設計に対する利用者の検査の時点を逃せば，システムに対するその後の変更要求には多大な経費がかかるものと覚悟しておかなければならない。

イ システムの概要設計書は，利用者が理解しやすいものでなければならないので，文章だけでなく適切な図表を併用することが望ましい。

ウ 大きなシステムでは，利用者側が設計書の詳細な内容を理解することは困難に近いので，設計書の検査は事実上，設計者側で責任を持つことになる。

エ 大きなシステムの場合には，設計書の内容に矛盾がないことを設計者が保証しなければならないが，設計書の最終的検査は利用者側の責任である。

オ　少数の一括処理プログラムからなるシステムでも，簡単な機能を表すために多量の設計文書が必要になることがある。

（特種　昭和55年）

【重要語句】

▨ imperative　避けられない　▨ therefore　それゆえ
▨ straightforward　率直に　▨ improbable　ありそうもない

【解説】　システム設計の初期の段階で必要となるユーザ側の仕様書の点検，確認の重要性に関して一般的意見を述べている。

以下に不適切な選択肢について理由を示す。

イ 「文章だけではなく適切な図表を併用することが望ましい」のは一般的には事実ではあるけれども，本文にはこのような記述はない。

ウ 「設計書の検査は……もつことになる」と反対の内容が，The details of the design are obtained を含む文節に述べられている。

オ 「多量の設計文書が…」とはどこにも記述されていない。

【解答】 ア　エ

［メ モ］

(62) シミュレーション

次の記述中の[ ]に入れるべき適当な字句を解答群の中から選べ。

(1) [a] is a technique of solving problems. In the broadest sense, it is a form of imitation in which a problem that needs solving is represented by a model which, in effect, replaces the problem by a second problem that is easier to solve.

(2) [a] replaces a system to be studied by some other representation called a [b]

(3) A [c] is described by algebraic and differential equations in which variables represent the attributes of the entities, and functions represent the activities.

(4) A [d] takes the form, principally, of a set of numbers that count the entities and indicate their status.

(5) Many [a] studies are concerned with [e], which is designed to understand an existing system with a view, perhaps, to changing a policy or procedure in order to modify the system performance.

解答群

ア system　イ discrete system　ウ system analysis
エ system design　オ simulation　カ discrete system simulation
キ continuous system simulation　ク model　ケ discrete model
コ continuous model　(特種 昭和55年)

【重要語句】

- in the broadest sense 広義では
- in effect 実際には
- algebraic equations 代数方程式
- variables 変数
- differential equations 微分方程式
- functions 関数

【解説】 シミュレーションに関する基本的知識を部分的に述べている英文である。シミュレーションとは，あるシステムの性能などを評価するために，その模型（モデルといわれる）を用いて実験を行うことをいう。

以下，空欄ごとに，適当な単語を検討する。

(1) 現実の状態をもっと解きやすいものに置き換える（by a second problem

that is easier to solve）からシミュレーションの定義であることが分る。現実の状態を1次的というのに対し，その特徴を生かした模型の状態を2次的という。

(2)　他の表現で置き換えたものを model という。

(3)　システムが関数で表現できると述べている。すなわち，モデルが代数方程式や微分方程式などの連続関数で表現できるということであるから continuous model（連続モデル）が該当する。

(4)「ある値を与えることによって，1組の数値が得られる」ということである。すなわち，モデルが関数として表現できないことを示しているから discrete model（不連続モデル）が該当する。

(5)　シミュレーションを行うために必要な事項について述べている。e に該当する語は，実在のシステム（existing system）を理解するために必要なものということである。すなわち，system analysis（システム分析）が該当する。

【解答】 a. オ　b. ク　c. コ　d. ケ　e. ウ

［メ　モ］

第7章 その 他

(63) 計算機用語—2—

次の各文aからeはそれぞれ何について説明したものであるか，解答群の中から適当なものを選べ。

a　Designed to be a common business language for commercial users to be implemented by several computer manufacturers.

b　A storage device used to compensate for a difference in rate of flow of data or in time of occurrence of events when transmitting data from one device to another.

c　Pertaining to peripheral equipments or devices in direct communication with the central processing unit of a computer.

d　The unit of a system that contains the circuits that control and perform the execution of instructions.

e　The totality of programs and routines used to extend the capabilities of computers, such as compilers, assemblers, and subroutines.

解答群

ア　central processing unit (CPU)　　イ　software　　ウ　COBOL

エ　buffer　　オ　on-line　　(第2種　昭和47年)

【重要語句】

- storage device　記憶装置
- circuit　回路
- instruction　命令
- peripheral　周辺の

【解説】 以下に解答群の用語について説明する。

central processing unit(CPU)　中央処理装置と訳すが，一般的にはCPUと呼ばれている。これは，コンピュータの演算，制御などを行う部分と，プログラムやデータなどを記憶する主記憶装置（main memory）とからなる。いわゆるコンピュータ本体と呼ばれるもので，LSIなどの回路素子や磁心記憶装置などが収容されている。なおd中のsystemは，主として，ハードウェアに関するコンピュータシステム（電子計算機組織）を指している。

software　利用技術とも訳されるが，一般的にはソフト（ウェア）という。こ

れは，狭義ではコンピュータの能力を最大に発揮させるためのプログラム群（system program）のことであり，オペレーティングシステム（operating system：OS）と同義に用いられる。またソフトウェアが拡大解釈されて，ハードウェアに属さないシステム・サービス全般を指して使われることもある。例えば，処理方式などが該当する。一方，ハードウェアとは，コンピュータシステムの電気的物理的なものを指す。

COBOL　COmmon Business Oriented Language の略。事務用のデータ処理に用いられる共通標準プログラム言語で，コンピュータメーカから提供される。

buffer　緩衝記憶装置ともいわれる。2個の装置間で動作の歩調が違うとき，その中間に設けて，速度，時間などの調整を行う。例えば，入出力装置のように動作速度の遅い装置と内部記憶装置の間に設けられる。

on-line　コンピュータの CPU 制御のもとにある周辺装置の結合状態，あるいは，ある機器とある機器がライン（伝送回線）で結合されている状態のことをいう。なお，c では装置，機器という意味で異なった単語が用いられており，多少のニュアンスの差はあるが，慣例的な用い方であることが多い。以下に，意味を示しておく。

equipment（装置，機器）　peripheral equipment（周辺装置）

device（装置，機構）　unit（装置，単位）

【解答】 a. ウ　b. エ　c. オ　d. ア　e. イ

［メモ］

(64)　数の表現

次の英文中の［　　］に入れるべき適当な単語を解答群の中から選べ。

A number may assume various symbolic [a]. In natural languages, the English "three" is definitely different from the French "trois," even though both represent the same [b]. Mathematically, the same [b] can be expressed as "3" in [c] notation, as "11" in [d] notation, and so on. In machine design we are primarily concerned with the expense of implementing a given number representation and with how well arithmetic can be performed with that representation. In particular, our interest is restricted to a number system with [e] representation because of the physical [f] of machines. As a result, underflows, overflows, scaling, precision, and [g] error characterize the machine number system and, consequently, its arithmetic.

解答群

ア	round-off	イ	integer	ウ	decimal
エ	binary	オ	quality	カ	quantity
キ	conditions	ク	limitations	ケ	finite
コ	representations				

（第2種　昭和52年）

【重要語句】

- even though～　　たとえ～しても
- and so on　　など
- notation　　表記法（decimal notation：10進法，binary notation：2進法）
- be concerned with～　　～にかかわる
- as a result　　結果として
- because of～　　～のために
- consequently　　結果として

【解説】　コンピュータで取扱う数字体系についての記述である。前半では量（質ではない）を表現するのに，自然言語では種々の方法（例えば3を表現す

るのに英語では three, フランス語では trois) があるが数字でも同じことがいえる (10進法では3, 2進法では11) と説明されている。後半では, 限られた数体系と, コンピュータのハードウェア面からくる制限 (physical limitations) 及び費用面からの制限などを考慮して演算機構が設計される。それゆえ, 演算機構がそのコンピュータを特徴づけると論述されている。

この演算機構を特徴づける要素として, underflow などが並列的に挙げられているので, これらについて以下に簡単に説明する。

overflow, underflow 演算結果が, コンピュータの扱える数の範囲を超えた場合を overflow (オーバーフロー) またはあふれという。また, 逆に下まわったときを特別に underflow (アンダーフロー) という。オーバーフロー, アンダーフローは, あふれとしてまとめて呼ばれることが多い。

scaling 固定小数点演算において, 数値が, コンピュータの扱える範囲の値となるように位取りすること。スケーリングとそのまま呼ばれる。

precision 精度と訳される。

round-off 丸めと訳される。無限小数を有限桁数として計算する場合の下の桁を落とすことなどをいう。

【解答】 a. コ　b. カ　c. ウ　d. エ　e. ケ　f. ク　g. ア

[メ モ]

(65)　正数，負数の表現

次の英文中の［　　］に入れるべき適当な字句を解答群の中から選べ。

In computers, negative numbers may be represented in either the sign and [a] form or the [b] form. The basic idea in the former representation is as follows : Assume that n bits are enough for representing all the positive numbers ([c] different configurations) to be covered. Then the addition of [d] to the original n bits will double the capacity for representing numbers. As a consequence, this newly acquired capacity can be assigned to represent [e] numbers.

解答群

ア　2n	イ　2^n	ウ　one bit
エ　two bits	オ　positive	カ　negative
キ　mantissa	ク　magnitude	ケ　component
コ　complement		

(第2種　昭和53年)

【重要語句】

- mantissa　（対数の）仮数
- magnitude　大きさ
- component　構成要素
- complement　補数

【解説】 以下，文章を順に追って説明する。

a，b　計算機内部での数値の表現方法には2通りある。1つは，符号ビットに続けて，正数を与える表現であり，他方は，補数を用いる表現である。

c　n ビットあれば，$0 \sim 2^n - 1$ まで 2^n 個の数字が表現できる。

d　$(n+1)$ビットになれば，$0 \sim 2^{n+1} - 1$ まで 2^{n+1} 個の数字が表現でき，これは，表現できる数字の範囲が2倍になる。すなわち，one bit が該当する。

e　前後関係から判断すると，「数の範囲が2倍になったため，そこに新しい表現を割り当てる」というような意味になる。また，この英文全体が負数表現について述べていることから，新しく追加された範囲を負数に割り当てると解釈するのが適当であろう。従って，negative が該当する。

なお，この英文では，sign and positive form についてのみ説明されているので，以下に complement form について説明しておく。

与えられた数をある特定の数から引いて得た数のことで，与えられた数の基数をxとすると，その数に対して，「x の補数」と「$(x-1)$ の補数」がある。計算機の内部では2進法が用いられるのが一般的であり，基数が2であるから「2の補数」と「1の補数」がある。いま，n 桁の2進数をbとすると「2の補数」C_2 は(2^n-b)で示される。2^n は$(n+1)$ 桁の2進数 $(\overbrace{100\cdots\cdots0}^{n\text{桁の}0})_2$であるが，$2^n$ の $(n+1)$ 桁目を無視，すなわちn桁全てを0とすると $C_2=-b$ となり，b の「2の補数」C_2 は $-b$ に等しくなる。$(n+1)$ 桁目を無視した演算を考えると，減算は，減数を加算すれば良いことになる。例えば，(3−2)を計算するとき，3+（2の2の補数）と考えれば，全て加算で処理が行える。

一方，b の「1の補数」C_1 は，「2の補数」から1を引いたものであり，(2^n-1-b) となる。(2^n-1) は，n 桁の2進数の最大値 $(\overbrace{11\cdots\cdots1}^{n\text{桁の}1})$ である。従って，$C_2=C_1+1$ となる。例えば，$n=4$ のとき $b=(0101)_2=$(10進数で5)としたとき，$C_1=(2^4-1-5)=10$。2進数で表現すると $(1010)_2$である。すなわち，$(0101)_2$ に対する C_1 は $(0101)_2$ の各ビットを論理否定（ビットの内容を反転）したものとなる。C_2 は $C_2=C_1+1$ であるから $(1011)_2$ となる。上記の (3−2) を例にとって加算例を示しておく。

```
10進数 2………0010                  0011…… 3
 2の1の補数…1101        ┌──→+) 1110…… 2の2補数
 2の2  〃   …1110───┘       10001…… 1 ──→(3−2=1)
                              ↑
                           無視する
```

【解答】 a. オ　　b. コ　　c. イ　　d. ウ　　e. カ

［メ モ］

(66) 計算機用語—3—

次の英文のうち内容の正しいものを四つ選べ。

ア　A bit is the smallest unit of information in the binary number system.

イ　A program is a series of instructions and commands for the computer to follow.

ウ　Batch processing is used in time-sharing system.

エ　Assembly language coding is absolute coding.

オ　An on-line printer operates under control of the central processing unit.

カ　I/O devices are high-speed devices, compared with the speed of central processors.

キ　An operating system increases the efficiency and usefulness of computer hardware and simplifies the programming job.

ク　In general, COBOL is used for writing programs designed to solve scientific or mathematical problems.　　(第2種　昭和53年)

【重要語句】

- binary number system　　2進法
- compared with～　　～と比較して

【解説】

ア　2進法は，基数に0と1の2個の数を用いる位取り基数法である。すなわち，2つだけの数字で全ての値を表現する。このとき，0は最低位の数字で1は最高位の数字である。2進法の1桁を bit（ビット）と呼ぶ。これは，binay digit の略である。なお，2進法は，binary notation ともいう。

イ　コンピュータは，人間が与えた処理手順に従って処理を行う。この人間が与える処理手順のことをプログラムといい，プログラムを作成することをプログラミングという。

ウ　Batch processing はバッチ処理と呼ばれ，事象の発生と同時に処理を行うリアルタイム（realtime）処理に対して，事象の発生後中間媒体（カード，テープなど）によって一定期間まとめて処理を行う方式のことをいう。また

time-sharing system は，TSS と呼ばれ，一定時間内に複数のジョブがコンピュータを共有する方式のことである。時分割処理ともいう。この方式によって，端末を用いているユーザが，同時に1台のコンピュータをあたかも自分1人で使用しているかのように使えるようになった。

エ　絶対コーディングとは，絶対番地（absolute address）を用いたプログラムのコーディングのことであるが，これは，Assembly language ではなく，machine language（機械語）によるコーディングのことをいう。

オ　中央処理装置（central processing uint : CPU）と結合されている印刷装置であるから，当然，CPU の制御下で作動する。ある程度の処理機能（intelligent）をもつ装置であれば，CPU の制御を受けない場合もある。

カ　一般的には逆である。CPU の処理速度の方が速いため，入出力装置からデータを受け取ったり，書き出したりする場合，バッファリングなどの手法が必要となる。

キ　オペレーティングシステムの目的は，システム資源の有効利用とプログラミングなどソフトウェアの開発負担を軽減することである。

ク　これは，FORTRAN の機能である。COBOL は，事務処理用の言語である。

【解答】 ア　イ　オ　キ

［メ モ］

(67) 計算機システム

次の情報処理システム及びコンピュータに関連する英単語群中から異質のものを一つ選べ。

a　These words are names of media used for data processing systems.

ア　paper tape　　イ　punched card　　ウ　magnetic tape
エ　floppy disk　　オ　mark sheet　　カ　mark card reader

b　These words are names of various kinds of online processing systems.

ア　remote batch system　　イ　realtime processing system
ウ　distributed database　　エ　distributed magnetic tape library
オ　time sharing system　　カ　remote inquiry system

c　These words are names of elements used for computer memory.

ア　terminal　　イ　IC　　ウ　magnetic core
エ　thin film　　オ　transistor　　カ　electron tube

d　These words are names of famous computers in the early age.

ア　EDSAC　　イ　ENIAC　　ウ　EDVAC
エ　AMDAHL　　オ　UNIVAC-I　　カ　ASCC-MARK-I

e　These words are names used at flow-charting.

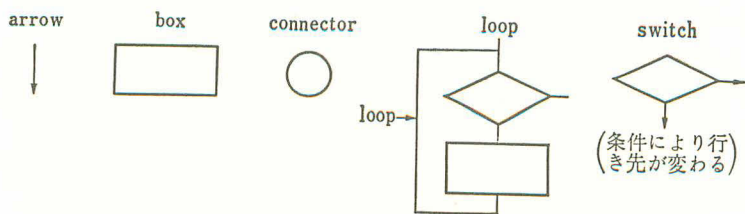
ア　arrow　　イ　box　　ウ　connector
エ　loop　　オ　queue　　カ　switch

f　These words are names of various types of files.

ア　sequential　　イ　random　　ウ　dialogue
エ　directory　　オ　index sequential　　カ　ring

(第2種　昭和54年)

【解説】　dの省略名は第Ⅲ部 略語辞書編に示した。掲載していなかったAMDAHL及びeの流れ図記号について説明する。

AMDAHL　G.M. Amdahl が IBM を退社して作った電子計算機会社。

【解答】 a. カ　b. エ　c. ア　d. エ　e. オ　f. ウ

〔メ　モ〕

(68)　計算機用語―4―

次の単語またはその解説文中の[　　]に入れるべき適当な字句を解答群の中から選べ。

1 indirect address : An address that specifies a [a] that contains either direct address or another indirect address.

2 [b] : Pertaining to a process or system that can be augmented.

3 relative address : The number that specifies the difference between the [c] and the base address.

4 on-line : Pertaining to equipment or devices under [d] of the central processing unit.

解答群

ア open-ended　　イ direct control　　ウ virtual memory

エ storage location　　オ absolute address　　カ network

(第1種　昭和47年)

【重要語句】

▨ pertaining to～　　～に関係すること

▨ difference　　差

【解説】 以下に，計算機用語のうち，重要なものについて説明する。

open-ended　新しい項目を追加しても，システム全体の構成を変更する必要のないことをいう。計算機の設計などでいえば，将来開発が予想される装置に対して，内部回路やプログラムの構成を全面的に変更することなく受け入れることができるように作成されていること。このように設計することをopen-ended design という。

base address　ベースアドレスあるいは基底アドレスと訳す。これはプログラムが主記憶装置にロード（格納）されたとき，占有した記憶装置の先頭のアドレスのことをいう。このアドレスは，絶対番地である。

absolute address　絶対番地あるいは絶対アドレスと訳す。これは，記憶装置にあらかじめ与えられた番地のことである。

relative address　相対番地あるいは相対アドレスと訳す。別に指定されるアドレスを基準として相対的に表わされたアドレス。基準アドレスとしては，

通常ベースアドレスが用いられる。

virtual memory　仮想記憶（装置）あるいはバーチャルメモリと訳す。主記憶装置と補助メモリ（磁気ディスク装置，磁気ドラムなど）を見掛け上一体とし，必要な情報が主記憶装置にない場合は補助メモリから，ハードウェアあるいはソフトウェアの手段によって，自動的に主記憶装置に必要な情報が転送される。プログラマから見れば，見掛け上補助メモリと主記憶装置を合わせた分だけの大容量メモリが与えられる。

【解答】　a. エ　　b. ア　　c. オ　　d. イ

〔メ　モ〕

(69) プログラム及びサブルーチン

下記の説明文に該当する語句を解答群から選べ。

a　A small set of instructions in a main program for temporary transfer of control to a closed subroutine.

b　To print out, for program checkout, contents of machine registers and/or specified memory locations during program execution.

c　Special mechanisms included in central processing units which enable computers to run programs originally written for different type of machines.

d　A system having multiple central processing units so that multiple number of programs may be in execution concurrently within the system.

e　A programming technique employed when total memory requirement for a program is greater than physically available size.

解答群

ア　multiprogramming system　　イ　storage allocation
ウ　overlay　　エ　multiprocessor systems
オ　procedure　　カ　emulator
キ　calling sequence　　ク　snapshot

(第1種　昭和48年)

【重要語句】

- so that　～するために
- concurrently　同時に

【解説】 以下に，問題文の closed subroutine 及び解答群で選択されない語句について説明する。

closed subroutine　閉じたサブルーチン。一連のプログラムルーチンとして主ルーチンに直接組み込まれるサブルーチンではなく，主ルーチンとリンケージエディタを通して組み立てられるサブルーチン。closed subroutine に対して主ルーチンに直接組み込まれるサブルーチンを open subroutine（開いたサブルーチン）という。

multiprogramming system　1つのコンピュータシステムで見掛け上2つ以

上のプログラムを同時に実行することのできるシステム。同時とはいってもCPU は同時に2つの処理を行うことはできないので，1つのプログラムが終了してから他のプログラムに移るという逐次処理ではなく，複数のプログラムを準備しておき，これらを交互に実行することである。例えば，あるプログラムが入出力要求など CPU 以外の装置を使用しているとき，別なプログラムを CPU が処理するような方法である。

storage allocation データを格納したり，他の特別な要求のために記憶装置の領域を確保すること。

procedure データを操作したり問題を解いたりするための手順。手続きと訳す。

【解答】 a. キ　b. ク　c. カ　d. エ　e. ウ

〔メ モ〕

(70)　マイクロコンピュータの動向

次の英文の内容と一致する文章を解答群の中から二つ選べ。

Within the past two years, a new technical hobby has come into existence: microcomputers. It appears that there may be in excess of 20,000 computers in not-for-profit use in homes today. Until now most computers supplied in kit form required a significant familiarity with electronics. Today, however, we are seeing the introduction of pre-assembled, ready-to-use systems. This signals the movement from hobby computing into personal computing, in which the user is more interested in the uses of the machines than in their physical construction.

Hobby computing differs from computing in industry or business in several important respects. To the hobbyist, price is of major concern; speed and reliability are secondary. The users' time and effort may essentially be considered to be free. The primary motivation is entertainment in its broadest sense. Physical appearance of the systems is of only minor concern.

解答群

ア　この2か年間，本国で使われている 20,000 台以上のコンピュータは利益をあげていないように思われる。

イ　これまでのコンピュータ愛好家と最近の個人持ちコンピュータの利用者とでは興味の対象が若干ちがう。

ウ　愛好家によるコンピュータの利用と産業用のそれとは種々の点で違いが見られ，その最大の違いは速度と信頼性である。

エ　これまでキット売りコンピュータを使うには電子工学の十分な知識が不可欠とされていた。

オ　産業用コンピュータの利用者も本質的には自由であり，広い意味でコンピュータ利用を楽しみとしている。　(第1種　昭和52年)

【重要語句】

- not-for-profit　利潤追求をしない
- pre-assembled　前もって組み立てられている

▨ ready-to-use　すぐ使用できる

▨ respect　点

【解説】 以下に，解答群の選択肢について検討する。

ア 全訳から分るように，20,000台以上のコンピュータが利潤追求のため以外に家庭にあると述べられている。

イ これまでのコンピュータ愛好家は，エレクトロニクスの知識が必要で，どちらかというと，ハードウェア（physical construction）に興味をもっていたが，最近の個人用コンピュータは，機械を使うこと（the use of the machine）に興味をもっていると述べている。

ウ Hobby computing differs from 以降は，hobbyist（愛好家）と industry or business（産業）との比較を行っている。愛好家にとって価格が大きな関心事であり，スピード，性能は二の次である。

エ これと同じ記述が，4行目から5行目に書かれている。

オ 「産業用コンピュータの利用者も本質的に自由である」とはどこにも書かれていない。

【解答】 イ　エ

［メ モ］

(71) パーソナルコンピュータ

次の英文中の[ ]に入れるべき適当な字句を解答群の中から選べ。

The personal computer can be regarded as the newest example of human mediums of communication. Various means of storing, retrieving and manipulating information have been in existence since human beings began to talk; [a] mediums serve to capture [b] thoughts for communication and, through feedback process, to form the paths that thinking follows. Although digital computers were originally designed to do [c] operations, their ability to [d] the details of any descriptive model means that the computer, viewed as a medium, can simulate any other medium if the methods of simulation are sufficiently well described. Moreover, unlike conventional mediums, which are [e] in the sense that marks on paper, paint on canvas and television images do not change in response to the viewer's wishes, the computer medium is [f]; it can respond to queries and experiments and can even engage the user in a two-way conversation.

解答群

ア	modify	イ	internal	ウ	fast	エ	passive
オ	file	カ	simulate	キ	active	ク	slow
ケ	arithmetic	コ	external				

(第1種 昭和53年)

【重要語句】

- retrieve 検索する (retrieval・検索)
- manipulate 操作する
- medium 媒体 (複数形は media または mediums)
- human being 人, 人間

【解説】 コミュニケーションの手段としての個人用の計算機の意義に関して記述した英文である。communication という語は,「通信」と訳すより, この場合,「情報伝達」という訳の方が適当である。

以下空欄を含む文節ごとに検討する。

a, b 直訳すると [a] な媒体が, 情報伝達に関して [b] な思想を得

ることを助ける」となる。すなわち媒体と思想が対比されており，解答群から，適当な反意語をみつけると，internal と external がある。媒体は external であり，思想は internal である。

c，d 直訳すると，「ディジタルコンピュータは，もともと，[c]な操作を行うために設計されたものであるが，どのような記述モデルでも，詳細を[d]する能力は…ということを意味する」となる。「…」の部分は，that 以下の文節で，「コンピュータも1つの媒体とみなすならば，シミュレーションの方法についての詳細な記述があれば，どんな媒体でもシミュレーションできる」という内容である。c は，digital computer の歴史からいっても，arithmetic 以外適当な単語はない。また，d は，that 以下の文節で simulation について述べているのであるから，simulate が該当する。

e，f 「伝統的な情報伝達の媒体（紙，絵，テレビ）が自分で加工できない」のに対してコンピュータは自分の意志で操作できることをいっている。これも，a，b と同様対比的な表現になっていること，伝統的な媒体が自分が加工できないことから「受身である」という表現が適当である。従って，e は passive，f は active が該当する。

【解答】 a. コ　b. イ　c. ケ　d. カ　e. エ　f. キ

［メ モ］

(72) システム開発一般

次の英文中の[ ]に入れるべき適切な単語を解答群の中から選べ。

System development proceeds from project definition through design to implementation, test, production, and maintenance. However, many people have assumed that minor differences between software and hardware (such as a paper product instead of electronics or a single product instead of a manufacturing line) do not affect the [a] life cycle. In fact, these differences have a [b] impact on the relative importance of decisions made in the life cycle. Their effect is immediately evident when you draw a chart of the project cost as a function of time. Software projects [c] their peak expenses during implementation and test.

The most profitable hardware project is one which invests a reasonable amount of resource in building an engineering prototype and then exercises and reworks it to find the least expensive way to manufacture copies. This [d] engineering phase occurs at the end of the prototype development and test phase. If it is done well, the instructions to the plant will call for the fewest, lowest-cost parts and the simplest, least expensive production processes necessary to [e] a machine with all the required functions and features. In other words, by reworking the prototype to define the lowest unit cost of production, the manufacturing cost and thus, the total project cost will be minimized.

解答群

ア product	イ research	ウ systems
エ development	オ assemble	カ negligible
キ market	ク incur	ケ avoid
コ significant		

(特種 昭和54年改題)

【解説】 同じシステム開発でも，ソフトウェアとハードウェアではその過程が基本的に異なるのであるということを，例を挙げて述べている。

以下，空欄ごとに適当な語を検討する。

a　ソフトウェアとハードウェアのシステムの開発のライフサイクルが同一に

考えられやすいことが述べられている。development（開発）と system（システム）どちらでも意味的には問題はないが，開発ということが主である。

b　ソフトウェアとハードウェアは，各時点で異なった開発をする必要があるが，同じようにするとインパクトが起きるという主旨である。インパクトを形容する単語としては，significant（重要な）が適当であろう。

c　ソフトウェア開発で最も費用がかかるときのことをいっている。従って incur（招く，こうむる）が適当である。

d　ハードウェア生産過程の手法が記述されており，これを示す語が必要である。従って，product（生産）が適当である。

c　装置などを実際に生産するときである。従って assemble（組み立てる）が適当である。

【解答】 a. エ　b. コ　c. ク　d. ア　e. オ

［メ モ］

(73)　データベース管理システム（DBMS）

データベース管理システム（DBMS）に関する次の英文の内容に最も近い文章を解答群の中から二つ選べ。

The continuing development of data base management systems (DBMS) is facing a severe challenge. As the capability of a DBMS increases, so does its overhead. A way must be found to reduce this overhead without reducing—indeed while expanding—the capabilities of current systems. The use of back-end processors to solve the problem would be similar to the use of front-end processors for handling data communications. As users required increasingly complex communications and message processing capability, the message processing function was put on a separate processor. Now data base management systems have approached a comparable stage of development and also might efficiently be offloaded to a special processor. Traditionally, many other computing functions began as individual programs as well, moved to general purpose packages, and eventually into hardware. This was true of floating-point arithmetic, index registers, and virtual memory, among others. One should expect the same thing to occur with DBMS functions.

解答群

ア　DBMS に含まれるデータ通信機能を専用のプロセッサに肩替りさせることが，今日の DBMS の難点を解決する一つの方法である。

イ　DBMS の機能として必要になる浮動小数点演算，指標レジスタ，仮想記憶などの機能は，やがてハードウェアによって実現されるようになる。

ウ　ここで考えているプロセッサは，DBMS 専用のプロセッサである。

エ　フロントエンドプロセッサは，データ通信のためのメッセージ処理専用のプロセッサである。

（特種　昭和54年改題）

【重要語句】

- processor　プロセッサ（処理装置）
- eventually　最終的に

▩ floating-point　浮動小数点

【解説】 データベース管理システム（data base management system：DB-MS）における専用処理装置の必要性をデータ通信のフロントエンドプロセッサと対比して述べた英文である。DBMS では，データ通信のフロント，エンドプロセッサに相当する処理装置をバックエンドプロセッサと称している。フロントエンドとは，前置計算機と訳せるが，一般的にはフロントエンドという。これは，「システムを効率よく働かせるため，ホストコンピュータとは別に設置される処理装置」のことで，このフロントエンドは，入出力制御や通信制御などを行い，クリーンデータのみをホストコンピュータに送る。このようにすることにより，ホストコンピュータの負荷が軽減する。

以下，各選択肢のうち不適当なものについて理由を示す。

ア　DBMS に通信機能があるという記述は英文中にはない。専用のプロセッサに肩替りさせる目的は，オーバーヘッドを減少させるためと述べている。

イ　これに関する記述は本文中にはない。

【解答】 ウ　エ

［メモ］

第Ⅲ部　略語辞書編

コンピュータ関係の書物には，英単語の省略形が頻繁に出てくる。省略形は一般の辞書にはよほどのことがない限り掲載されないことが多い。またコンピュータの用語辞典でも五十音順であるから，例えば「ISO」を「アイエスオー」と読むか「イソ」と読むか分らないとき，複数の個所をさがさなければならない。また読み方が分らないと全くさがせないときもある。

この略語辞書編では略語を「一般用語」，「言語及び汎用パッケージ」，「団体及び機関名」に分け，さがし易さを目標に，省略形をアルファベット順に並べ，正確な綴り，読み方を示し，その内容を簡単に説明した。さらに詳しい内容を知りたい場合は，専門の辞典・便覧などを参照して頂きたい。実際の英文では，団体・機関名などは略語で示すことが多いが，慣れてくれば英文の前後関係から判断できる。団体・機関名か否かはっきりしない場合は一般と団体両方を参照して欲しい。

〔1〕 一 般 用 語

〔A〕

A.C. (Alternating Current) 交流。

ADA 米国防総省が開発した COBOL, FORTRAN にかわる新しい言語で，信頼性，保守性の向上に重点がおかれている。

ADP (Automatic Data Processing) 自動計算機によりデータ処理を行うこと。

AM (Amplitude Modulation) 振幅変調

ANSCII (American National Standard Code for Information Interchange) 米国国家規格協会 (ANSI) で定めたパリティビット (奇偶検査ビット) を含めた8ビットの情報交換用符号。

ASCC (Automatic Sequence Controlled Calculator Harvard MARK-I) Babbege の考案した Analytic Engine (解析エンジン) の理論を実現した初期のリレー計算機。Harvard 大学の H.Aiken と IBM 社の協力により，1944年に完成した。なお ASCC は，Harvard MARK-I ともいう。

〔B〕

B-register index register と同義。指標レジスタ。

BBD (Bucket Bridge Device) バケットブリゲード素子。CCD(Charge Coupled Device) と類似の半導体素子で，二相駆動が可能であるが，速度は CCD よりも劣る。

BCD (Binary-Coded Decimal) 2進化10進。10進数の各桁を2進法によって表現したもの。

BE-MOS (Beam Addressed MOS) 大容量，高速で安価な記憶装置。CRT (Cathode-Ray Tube: 陰極管) 上のシリコンターゲットに電子をあてて記憶する。

BOT (Beginning of Tape marker) テープの始端。

BPI(Byte Per Inch) 1インチ当りバイト数。

BPS (Bit Per Second または Byte Per Second) ビット/秒，バイト/秒。

〔C〕

C-MOS (Complementary MOS) 消費電力が少なく，高速性のある MOS。

CAD (Computer Aided Design) コンピュータを利用した会話型の graphic を用いた設計システム。

CAE (Computer Aided Engineering) CAD より広い概念であり，設計の全体を網羅する手法。

CAI (Computer Assisted Instruction) コンピュータを用いた会話型の個人教授システム。

CAM (Computer Aided Manufacturing) コンピュータによる製造業務の自動化システム。

CCD (Charge Coupled Device) 電荷結合素子。半導体の表面を酸化膜の絶縁物でおおい，その上に輸送電極を形成した単純な構造の MOS トランジスタ類似の素子。トランジスタに比べて集積度は高い。

CCE (Communication Control Equipment) 通信制御装置，データ通信において，通信回線と端末装置，中央処理装置の間にあって両者間の情報の授受に関する種々の制御を行う。

CCU (Communication Control Unit) 回線制御装置。CCEの主要装置で情報処理装置（中央処理装置，端末装置など）の側に位置する装置。あるいは，CCEと同義に用いられる場合もある。

CE (Customer Engineer) 計算機の整備や修理などをして保守する人のこと。

CIM (Computer Input Microfilm) マイクロフィルムに記録されている情報を直接計算機に入力する装置。

C³L (Complementary Constant Current Logic) 高密度，高速度のバイポーラ論理素子。

CMI (Computer Managed Instruction) 計算機を利用した教授管理システム。

COM (Computer Output Microfilm) コンピュータ出力を文字情報として直接マイクロフィルムにする装置。

CPM (Critical Path Method) ORの手法で，手順を矢線図で表わし作業の費用と所要時間の関係を直線で近似し，LP(線形計画法)を用いて最小の日程計画を求めるもの。

CPS(Character Per Second) 字/秒

CPU(Central Processer Unit) 中央処理装置

CRC (Cyclic Redundancy Check) 巡回冗長検査。比較的高密度で磁気テープに記録された情報の読取り誤りの発生を発見するための冗長検査法の一種。

CRT (Cathode-Ray Tube) 陰極線管。CRT Display とは CRT を用いた表示装置。

CTD(Charge Transfer Device) CCDと同じ。

CVCF (Constant Voltage and Constant Frequency) 定電圧定周波電源装置。

[D]

D-MOS(Depletion MOS) デプレッション型 MOS。

DASD (Direct Address Storage Device) 大容量の補助記憶装置。

DBMS(Data Base Management System) データベース管理システム。データベース管理のための汎用ソフトウェア。

D.C.(Direct Current) 直流。

DCTL(Direct Coupled Transistor Logic) 直接結合トランジスタ論理。構成が簡単で高速であり消費電力が少ないなどの利点があるが，トランジスタ，抵抗などの特性に対する要求がきびしく，温度によって変化し易く，雑音裕度が小さいなどの欠点がある。

DDA(Digital Differential Analyzer) 計数型微分解析器。微分方程式を解くために設計された微分解析器の精度を上げるために演算をディジタルな数値におきかえて行うもの。

DDC(Direct Digital Control) 直接制御ディジタル計算機によるプロセス制御の一種。

[E]

E-MOS (Enhancement MOS) エンハンスメント型 MOS。

EA-ROM(Erasable and Alterative Read-Only Memory) 一度書き込まれた記憶内容を電気的方法で消去し，再書き込みを行える不揮発性 ROM。

EBCDIC (Extended Binary-Coded Decimal Interchange Code) 拡張2進化10進コード。

ED-MOS (Enhance-Depletion MOS) D-MOS を改良したもの。

EDP(Electronic Data Processing) 主として計算機を用いて行われるデータ処理。

EDPS (Electronic Data Processing System) EDP システム。

EDSAC(Electronic Delay Storage Automatic Calculator) 1949年, ケンブリッジ大学の M.V. Wilkes を中心として開発された史上初のプログラム内蔵方式の計算機。

EDVAC (Electronic Discrete Variable Automatic Computer) von Neumann の提言を取り入れ, 1951年, ペンシルバニア大学で完成した計算機。

ENIAC(Electronic Numerical Integrator and Computer) ペンシルバニア大学の J.P. Eckert 及び J.W. Mauchly によって1946年に完成した真空管を用いた世界最初の電子計算機。

EOF (End of File) ファイルの終り。

EOT (End of Tape marker) テープの終端。

[F]

FIFO (First-In First-Out) 先に発生したものを先に取り出して処理をする考え方

FM (Frequency Modulation) 周波数変調。

[G]

GCR (Group Coded Recording) 磁気テープの記録方式の1つ。高密度化が可能である。

GSI (Grand Scale Integration) 超大規模集積回路。SLSI (Super Large Scale Integration) ともいう。

[H]

Harvard MARK-I ASCC のこと

HSP (High-Speed Printer) 高速印字装置。

[I]

IBG (Inter Block Gap) ブロック間のギャップ。何も記録されていない部分。

IC (Integrated Circuit) 集積回路。2個以上の素子が1つの固体内に収められている回路。軽量, 小型で消費電力が小さく, また, 同一規格のものを大量生産するのに適している。

IDP (Integrated Data Processing) 集中データ処理。

I²L (Integrated Injection Logic) 集積度が高く, 高速度, 低消費電力のバイポーラ論理回路の一種。

IOCS (Input Output Control System) 入出力制御システム。OS の中の入出力部分を管理するシステム。

IPL (Initial Program Loading) ハードウェアのロードボタンによる制御プログラムのローディング。

IR (Infomation Retrieval) 情報検索。

IRG (Inter Record Gap) レコード間のギャップ。レコード間の何も記録されていない部分。

ISCII (ISO Coded for Information Interchange) ISO によって定められた情報交換用符号。

ISO/DIS (ISO/Draft International Standard) ISO 国際規格案。

ISO/DR (ISO/Draft Recommendation) ISO 推選規格案。

ISO/IS (ISO/International Standards) ISO 国際規格。

[J]

JIS (Japan Industrial Standards) 日本工業規格。

JISC (Japan Industrial Standards Committee) 日本工業標準調査会。

[K]

K (Kilo) 1000 を意味する場合と 1024 (2^{10}) を意味する場合があるが, 計算機の記憶容量を示すとき (例えば 128K

Byte) は，128×1024 Byte を示すことが多い。

KWIC (Keywords In Context) 情報検索システムで使われる手法の一種。

[L]

LASER (Light Amplification by Stimulated Emission of Radiation) 光レーザ。レーザプリンタ，光通信などに利用されている。

LED (Light Emitting Diode) 発光ダイオード。

LP (Linear Programming) OR (Operations Research) の手法の1つ。目的関数が1次関数で，制約が1次不等式または等式であり，各変数は非負である。

LPM (Line Per Minute) 1分当り印字行数。印刷装置の性能を示す。

LSI (Large Scale Integration) 大規模集積回路。はっきりした定義はないが，現在，10^3～10^4 前後のゲート回路などが1つの基盤上に組み込まれている。

[M]

MASK-ROM (Masked Read-Only Memory) 製造工程において記憶内容を書いてしまう ROM。

MEDLARS (Medical Literature and Retreival System) 医薬文献検索システム。

MICR (Magnetic Ink Character Reader または，Magnetic Ink Character Recognition) 磁気インク文字読取装置または磁気インク文字認識。

MIS (Management Information System) 経営情報システム。

MODEM (Modulator/Demodulator) データ伝送用の変復調装置。

MOS (Metal Oxide Semiconductor) トランジスタの一種。MOS 型はバイポーラ型に比べて入力インピーダンスが高い，出力が大きい，消費電力が少ない，集積度が高いなどの特徴がある。

MTBF (Mean Time Between Failure) 平均故障間隔。相異なる故障間の動作時間の平均値。

MTTF (Mean Time To Failure) 故障までの平均時間。

MTTR (Mean Time To Repair) 平均故障修復時間。

[N]

N-MOS (N-channel MOS) N 型 MOS。

NC (Numerical Control) 数値制御。

NRZ (Non-Return-to-Zero) 磁気記録方式の一種。

[O]

OCR (Optical Character Reader または Optical Character Recognition) 光学文字読取装置または光学文字認識。

OMR (Optical Mark Reader または Optical Mark Recognition) 光学マーク読取装置または光学マーク認識。

OR (Operations Research) オペレーションズリサーチ。運営研究。体系の運営に関する問題に対して科学的な手法と用具（一般的には計算機）を適用して計量的に問題解決を計ること。

OS (Operating System) オペレーティングシステム。計算機を効率良く使うために用意された管理プログラム。

[P]

P-MOS (P-channel MOS) P 型 MOS。

P-ROM (Programmable ROM) 半固定記憶装置。使用者が電気的手段によって情報を書き込める ROM。

PABX (Private Automatic Branch Exchange) 自動式構内交換設備。

PBX (Private Branch Exchange)　構内交換機。

PCM (Pulse Code Modulation)　パルス符号変調。

PCS (Punch Card System または Print Contrast Signal)　パンチカードを媒体としてデータ処理を行うシステム。またはOCR用文字の印字品質を示す単位でPCS値という表現で用いる。

PERT(Program Evaluation and Review Technique)　仕事の手順，計画を矢線図で示し，時間的要素を中心に計画の評価，調整，進捗管理を行う手法。

PM (Phase Modulation)　位相変調。

POS (Point of Sales)　売場専用端末装置

PPBS (Planning Programming Budgeting System)　計画企画予算システム。

[R]

RAM (Random Access Memory)　等速呼出し記憶装置。アドレスに関係なく等速呼出しを行うことのできる記憶装置。

RIT (Rate of Information Throughput)　単位時間内に情報源から情報受信器へ伝達される情報量。

RJE (Remote Job Entry)　遠隔地の端末装置から通信回路線を通じてジョブを投入し，処理結果を端末装置から得る処理方式。

RO(Receive Only)　受信専用(端末装置)

ROM (Read-Only Memory)　固定記憶装置。記憶内容が回路上固定されており，自動的には書込みができない記憶装置。通常，定数や固定的なサブルーチンなどを記憶させておく。論理的には，磁気ドラムのような装置でも良いが，一般的には，半導体を用いた素子を指す。

RZ (Return-to-Zero)　磁気記録方式の1つ。

[S]

SCR(Silicon Controlled Rectifier)　サイリスタ（シリコンを用いたスイッチング素子）の一種。

SDI (Selective Dissemination of Information)　文献検索システムの文献情報を検索提供する方式の一種。文献情報がデータベースに登録されると，あらかじめ登録されている条件式(プロファイル)に応じて資格のあるユーザに対して自動的に情報を配布する方式。

SE (Systems Engineer)　問題を取り上げ，システム分析やシステム設計を行う人。

SMART (Salton's Magical Automatic Retrieval of Texts)　IBM 7094用の情報検索システム。

S/N (Single to Noise Ratio)　雑音に対する強さを示す尺度。S/N比が大きいと誤りの発生率が高くなる。

SOLOMON (Simultaneous Operation Linked Ordinal Modular Network)　ウェスティングハウス社で開発された並列計算機 (Parallel Computer)。

SOP (Study Organization Plan)　IBM社で開発した組織の研究と文書作成手法。

SSB (Single Side Band Transmission)　単側帯波伝送。

[T]

TSS (Time Sharing System)　時分割システム。一定の時間内に複数のジョブが計算機を共有し，あたかも，同時にジョブが実行されているようにするシステム

TTL (Transistor-Transistor Logic)　超小型ディジタル回路方式の一種。高速で消費電力が小さい。T^2Lと略すこともある。

TTY (Teletypewriter)　テレタイプライタ。送信側のタイプライタの操作により受信側のタイプライタが動作する方式または装置のこと。

[U]

UNIVAC-I (Universal Automatic Computer) 世界最初の商用計算機。

[V]

VLSI (Very Large Scale Integration) 超 LSI のこと。

VSM (Vestigial Side Band Modulation) 残留側帯波変調。

[W]

WAP (Work Analysis Program) 受注生産における工程を人員計画の面から管理する手法の1つ。

〔2〕 言語及び汎用パッケージ

〔A〕

ALGOL (Algorithmic Language)　科学技術計算用に開発された言語で，算法言語とも呼ばれている。

APL (A Program Language)　K.E. Iverson によって提唱された言語で，タイムシェアリング方式のコンピュータ利用のための会話型言語。強力な演算機能をもっている。

APT (Automatically Programmed Tools)　数値制御工作機械の制御テープを作り出す翻訳プログラム。部品の形状と工具の動作その他を記述した部品プログラムから工具軌跡を算出するプログラムが主要部分である。

〔B〕

BACCUS (Basic Calculus)　富士通で開発した会話型言語。

BASIC (Beginner's All Purpose Symbolic Instruction Code)　ダートマス大学の J.G. Kemeny, T.E. Kurtz が開発した FORTRAN 型の会話型言語。

BLISS (Basic Language for Implementation of System Software)　カーネギーメロン大学で開発されたシステム作成言語。structured programming（構造的プログラミング）の考えに基いたプログラム構造をもつ。

〔C〕

C FOR (Conversational FORTRAN)　ユニバックの会話型 FORTRAN。

COBOL (Common Business Oriented Language)　事務処理用に開発された共通言語。アメリカにおいて，コンピュータメーカと，ユーザからなる専門委員会 (CODASYL) によって開発された。

COGO (Coordinate Geometry)　土木工学における測量結果の面積計算を行う言語

CPL (Combined Program Language)　英国ケンブリッジ大学とロンドン大学により開発された言語で，非数値問題に対して ALGOL 型言語の適用範囲を拡大しようとした言語。

〔D〕

DETAB-X　CODASYL で提起された一種の非手順的言語。決定表 (decision table) 言語の母体となった実験的言語。

DYNAMO (Dynamic Models)　世の中の動きを連続的なものとしてとらえ，自動制御におけるフィードバック理論を基礎とした経済構造や，企業内組織内での各要素のからみ合いをときほぐそうという思想のもとに作られたシミュレーション言語。1961年 MIT（マサチューセッツ工科大学）で作られた。

〔E〕

EL1　ハーバード大学で開発されている言語で ALGOL 系の仕様をもつが，COBOL, LISP の概念も取り入れている。

〔F〕

FLOW-MATIC　G.M. Hopper を中心に1957年に完成した最初の事務計算向コンパイラ。UNIVAC I/II に用いた。

FORMAC (Formula Manipulation Compiler)　1964年 FORTRAN IV によって IBM 社により開発された数式処理言語。

FORTRAN (Formula Translation) 1956年に IBM 社により，IBM704 用に開発された科学技術計算用言語。その後拡張を重ね，今日では実用的に最も広く使用されている。

［G］

GIFS (Generalized Interrelated Flow Simulation) 化学工業におけるプロセスの分析を数学的モデルを使ってシミュレートするプログラム。

GPSS (General Purpose Systems Simulator) IBM 社の G. Gordon によって作られた待ち行列型のシミュレータ。

［I］

ICES (Integrated Civil Engineering System) 構造計算や橋りょう計算などに使われる言語。

IPL-V (Information Processing Language-V) リスト処理用言語であるが，LISP と比較すると機械語に近い。

［J］

JOSS (JOHNIAC Open-Shop System) 米国の RAND 社が開発したタイムシェアリング用の科学技術計算言語。JOHNIAC はプリンストン大学で開発された計算機の名称である。

［L］

LISP(List Processor) マサチューセッツ工科大学 (MIT) で開発されたリスト処理用言語。

［M］

MIDAS (Modified Integration Digital Analog Simulation) ライトパターソン空軍基地で開発されたディジタルシミュレーション言語。

MPS (Mathematical Programming System) 線形計画解析用のプログラムパッケージ。

［P］

PASCAL N. Wirth により提案された ALGOL に近い骨格の上に，構造的プログラミング (structured programming) の考え方を取り入れた言語。

PL/I (Program Language One) IBM社で開発された言語でプログラミング言語の中で最も高度といわれる。ALGOL のブロック構造，FORTRAN の簡略さ，COBOL のデータ構造など，他の高級言語の長所を取り入れ，初心者でもプログラムが簡単に作成できるような言語仕様となっている。

PLANNER 人工知能の領域の問題の記述を目的として MIT で開発された言語系。

［R］

RPG (Report Program Generator) 出力すべき報告書の形式，入力ファイルに関する情報などをパラメータとして与え，報告書を作成するプログラムを作成するプロセッサ。

RUSH (Remote Use of Shared Hardware) アレンバブコック社と IBM 社が共同で開発した会話型 PL/I。

［S］

SIMSCRIPT (Simulation Scriptor) RAND 社から発表された汎用シミュレーション言語。

SNOBOL ベル研究所から発表された記号処理用言語。

STRESS (Structure Engineering System Solver) 弾性構造体解析用の言語。

〔3〕 団体及び機関名

[A]

ACM (Association for Computing Machinery) アメリカ計算機協会。

AFIPS (American Federation for Information Processing Societies) アメリカ情報処理学会。

AMA (American Management Association) アメリカ経営協会。

AMS (American Mathematical Society) アメリカ数学学会。

ANSI(American National Standards Institute) 米国国家規格協会。

[C]

CBEMA(Computer and Business Equipment Manufacturers Association) 計算機事務機械工業会（米国）。

CCITT (International Telegraph and Telephone Consultive Committee) 国際電信電話諮問委員会。

CEE (International Commission on Rules for Approval Electrical Equipment) 国際電気機器承認規定委員会。

CEN (European Standards Coordinating Committee) 欧州規格調整委員会。

CENEL (European Electrical Standards Coordinating Commitee) 欧州電気規格調整委員会。

CODASYL(Conference on Data Systems Language) データシステム言語会議。

CODASYL DBTG (CODASYL Data Base Task Group) COBOL のデータベース管理機能を含める提案を行い，モデルを提示した委員会。

[E]

ECMA (European Computer Manufacturers Association) 欧州電子計算機製造工業会。

EIA (Electric Industries Association) 電子工業協会（米国）。

[I]

IBWM (The International Bureau of Weights and Measures) 国際度量衡局

ICOT (Institute for New Generation Computer Technology) （財）新世代コンピュータ技術開発機構。

IEC (International Electro-technical Commission) 国際電気標準会議。

IFIP (International Federation for Information Processing) 情報処理国際連合

ISO (International Organization for Standardization) 国際標準化機構。

ISO/TC 95 (ISO/Technical Committee 95) ISO の分科会。事務機械に関する標準化を行う。

ISO/TC 97 (ISO/Technical Committee 97) ISO の分科会。計算機と情報処理に関する標準化を行う。

ITU (International Telecommunication Union) 国際電気通信連合。

[J]

JEIDA (Japan Electronic Industry Development Association) （社)日本電子工業振興協会。

[O]

ORSA (Operations Research Society of America) 米国 OR 学会。

[P]

PASC(Pacific Area Standards Congress) 大平洋地域標準会議。

第Ⅳ部　JIS情報処理用語編

　ある程度英文解釈ができれば，分らない単語が出てきても，辞書を片手に英文を読むことができる。情報処理関係の書物についても同様である。ただしこの種の書物は文章が平易であり，技術用語（technical term）を除けば，一般の辞書はほとんど必要としないであろう。

　この用語編では，技術用語を集めて，和訳及び簡単な解説を行った。従ってコンピュータ用語の英和辞典として活用できる。本編は原則として JIS C 6230 情報処理用語に出典を求め，アルファベット順に並べた。JIS に示されている意味だけで分りにくいものには多少説明を加えた。用語中 (A)，(C)，(F) となるのは，ALGOL, COBOL, FORTRAN で用いられることを示す。なお PL/I は現在 JIS では定められていない。

（出典：JIS ハンドブック情報処理 1980，日本規格協会編）

［A］

absolute address（絶対アドレス） 主記憶装置に固有のアドレスであって，記憶場所を直接指定するアドレス。

access arm（アクセスアーム） 磁気ディスク記憶装置の一部分であって，一つ以上の読取り書込みヘッドを保持し，それらを所定のトラックに位置付けるためのもの。

access time（アクセス時間，呼出し時間） 制御装置が記憶装置からのまたは記憶装置へのデータ転送を要求してから，転送が完了するまでの時間。アクセス時間は待ち時間と転送時間とに分けられる。

accumulator（累算器，アキュムレータ） 演算装置にある主要なレジスタであって，四則演算，論理演算などの結果を保持するもの。多くの場合，累算器には一つの数値が保持されており，他から数値が入ってくると，両者の代数和でこれを置き換える。また，保持された数値について，けた送り，補数化などの操作もできるのが普通である。

actual argument（実引き数(F)） 手続きを使用する際に必要な値の受渡しをするのに用いられる文法単位。例えば定数，変数または式など。FORTRAN では，実引き数といい，手続きが引用されるとき，仮引き数と結合される。

actual parameter（実パラメタ(A)） 手続きを使用する際に必要な値の受渡しをするのに用いられる文法単位。例えば，定数，変数または式など。ALGOL では実パラメタといい，手続きが呼ばれるとき，仮パラメタとの対応がとられる。

adder（加算器） (a) 二つの数の和を作る回路。

(b) 3個の入力端子と2個の出力端子をもち，出力信号が入力信号に対し，次の表の関係にある回路。

表

入力			出力	
X	Y	C^*	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

X：被加数
Y：加　数
C^*：下位からのけた上げ
S：和
C：上位へのけた上げ

address（アドレス） 情報を転送する場合の出所または行き先を表す表示。例えば記憶装置の中で1語が占める特定の場所を指定するのに用いられる。

addressing（アドレッシング） selecting と同義。

address modification（アドレス変更） 命令のアドレスを変更すること。

address part（アドレス部） 命令の中でアドレスを指定している部分。

ADP automatic data processing の略。

ALGOL（アルゴル） 数値計算や論理演算を行うためのプログラム言語の一つ。

ALGOL は，Algorithmic Language の略。

algorithm（アルゴリズム，算法） 問題を解決するための明確に定義された有限個の規則または手順の集まりであって，それを有限回実行することによってその目的を達成することができるもの。

alphabetical character　letterと同義。

analog（アナログ，相似型）　データまたは物理量などを連続的に変化し得る物理量によって表現することを表す形容詞。

AND（論理積）　P及びQを二つの論理変数とするとき，次の表によって定める論理関数$P \cdot Q$を"PとQの論理積"という。$P \cdot Q$をPQ，$P \wedge Q$，$P \& Q$などと書くこともある。

表

P	Q	$P \cdot Q$
0	0	0
0	1	0
1	0	0
1	1	1

AND circuit（論理積回路，AND回路）　2個以上の入力端子と1個の出力端子とをもち，すべての入力端子に"1"が入力された場合にだけ，出力端子に"1"を出力する回路。

argument（引き数）　配列の要素を識別するのに用いる独立変数。

（引き数（F））　手続きの内容を定義したり，それを使用する際に必要な値の受渡しをしたりするために用いられる文法単位。FORTRANでは仮引き数と実引き数がある。

参考：ALGOLではparameterという。

arithmetic and logic unit（演算装置）　計算機の構成要素の一つであって，四則演算，論理演算などを行う装置。

arithmetic expression 算術式(A,F,C)　算術的な値が求められる文法単位。ALGOLでは，単純算術式と条件節を含む算術式とがある。単純算術式の基本的な構成要素には，定数，変数及び関数呼出しがある。条件節を含む算術式とは，次の形のものである。

"**if**〈論理式〉**then**〈単純算術式〉**else**〈算術式〉"

FORTRANの算術式の基本的な構成要素には，定数，変数名，配列要素名及び関数の引用がある。COBOLの算術式の基本的な構成要素には，数字基本項目の一意名及び数字定数がある。

arithmetic operator（算術作用素（A），算術演算子（F, C））　算術式を構成する要素。これらは，次の表のとおりである。

表

演　算	ALGOL	FORTRAN	COBOL
	算術作用素	算術演算子	算術演算子
加　算	+	+	+
減　算	−	−	−
乗　算	×	*	*
除　算	／，÷	／	／
べき乗	↑	**	**

備考：ALGOLの÷は結果が整数型となり，／は結果が実数型となる

array（配列（A，F））　一つ以上の次元に従って要素を並べたものと，同じ特性をもち，定められた規則に従って並べられた要素の集まりから成る文法単位。名前によって識別され，次元とその大きさとを指定することによって定まる。ALGOLでは，配列といい，添字付き変数で配列の要素を識別する。FORTRANでは，配列といい，配列名に添字を続けて書くことによって配列の要素を識別し，これを配列要素名という。

assembler（アセンブラ）　アセンブラ言語で書かれたプログラムを機械語のプログラムへ翻訳するプログラム。

assembler language（アセンブラ言語）　計算機向き言語の一つであって，通常，その各命令は計算機の命令と1対1の対応をもつ。マクロ命令のような機能を備えていることもある。

assignment statement（代入文（A，F））

実行時に右辺の式の値によって，左辺に示すものの値を置き換える働きをする文。ALGOLでは，右辺の算術式または論理式の値を，その左辺の変数や手続き名に与える働きをする。FORTRANでは指定された変数または配列要素に値を与えるもので，算術代入文，論理代入文及びASSIGN文がある。

association（結合（F）） 変数や配列要素が同一の記憶場所を共有することをいう。結合は，COMMON文，EQUIVALENCE文または引き数によって引き起こされる。

associative storage（連想記憶装置） 記憶場所がその内容によって識別される記憶装置。

asynchronous computer（非同期式計算機） 前に行われた事象や動作の終了を知らせる信号，または次の事象や動作に要する機構が使用可能になったことを知らせる信号によって，個々の事象や基本的動作の実行が開始される計算機。

asynchronous system（非同期式） 一つの文字またはブロック内では，どの二つの有意瞬間もその間隔が単位間隔の整数倍であるが，二つの文字またはブロック間では整数倍である必要のない同期の一形式。

attenuation distortion（減衰ひずみ） 所要周波数帯域内で，回線またはシステムの損失または利得が周波数に対して一定でないために起こるひずみ。

audio response unit（音声応答装置） 計算機などからの指令によって，応答内容を音声信号に変換して送り出す装置。

automatic coding（自動コーディング） 計算機によって原始プログラムを機械語のプログラムに変換すること。

automatic data processing（自動データ処理） 主として自動的手段によって行われるデータ処理。

automatic programming（自動プログラミング） プログラムを人間にわかりやすい形から，計算機が実行できるような形に計算機自身を用いて直す方法。自動プログラミングを行うには，コンパイラ，インタプリタなどが用いられる。

automatic repetition system（自動連送方式） データを一定時間の遅れをもって2回以上自動的に送り，誤りを検出する方式。

auxiliary storage（補助記憶装置） 主記憶装置の働きを補助するために用いられる記憶装置。

［B］

backward read（逆読み） 紙テープ読取り装置や磁気テープ装置などで，順方向とは逆の方向に媒体をもどしながら，記録されているデータを読み取ること。

band（バンド） 磁気ドラムなどにおいて，いつも同時に呼び出されるトラックの一組。

base radixと同義。

base address（基底アドレス，ベースアドレス） プログラムにおいて基準となるアドレスであって，これに相対アドレスを加えれば絶対アドレスが得られるもの。

basic external function（基本外部関数（F）） プログラマが定義しないで使うことのできる関数。例えば三角関数や対数関数などの初等関数。

basic statement（基本文（A）） 基本文には，代入文，飛越し文，空文及び手続き文がある。

basic symbol（基本記号（A）） プログラム言語で許される文字の集まり。次頁の表に示す文字がある。

a　b…z　　A　B…Z
0　1…9
true false
＋ － × ／ ÷ ↑ ＜ ≦ ＝ ≧ ＞ ≠ ≡ ⊃
∨ ∧ ¬
goto if then for do
, . ₁₀ : ; := △
step until while comment
( ) [ ] ' '
begin end own Boolean integer real array
switch procedure string label value

注：△は，"空白"を示す。

batch processing（一括処理，バッチ処理）　データ処理において，必要なデータをためておき，それをひとまとめに処理する方式。

baud（ボー）　変調速度の単位。

beginning of tape marker（テープ始端マーカ）　磁気テープ上の記録開始可能な位置を示す物理的な印。例えばアルミはくの反射マーカなどが該当する。

binary cell（2値素子）　二つの安定状態をもち，1ビットの情報を保持できる素子。

binary code（2進コード）　2値（"0"と"1"）を用いたコード。

binary-coded decimal notation（2進化10進法）　数の表記法の一つであって，10進法における各けたの10進数字を2進法で表すもの。

10進数	2進化10進法
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	0 0 0 1 0 0 0 0

binary digit（2進数字）　2進法の数字であって，0と1が用いられる。

binary notation（2進法）　2を基数とする固定基数表記法。数字は"0"と"1"を用いる。

binary point　radix point と同義。

binary search（二分探索）　項目の集まりを二つの部分に分け，そのどちらかを選び出す手続きを繰り返すことによって，求める項目を見い出す探索方法。通常項目は昇順または降順に並べられている。

biquinary notation（2-5進法）　符号化10進法の一種で，10進数字 n を $5\times n_1+n_2$ （$n_1=0,\ 1$；$n_2=0,\ 1,\ 2,\ 3,\ 4$）として表す表記法。2-5 進法は交互に5と2とを基数とする表記法ともみられる。

10進法	2-5 進法	解釈
0	01 00001	0+0
1	01 00010	0+1
2	01 00100	0+2
3	01 01000	0+3
4	01 10000	0+4
5	10 00001	5+0
6	10 00010	5+1
7	10 00100	5+2
8	10 01000	5+3
9	10 10000	5+4

bistable circuit（双安定回路）　二つの安定状態をもつトリガ回路。

bistable trigger circuit　bistable circuit と同義。

bit（ビット）　(a)　情報の量の単位であって，1個の2進数字の保有し得る最大情報量を表す。
　(b)　2進数字に同じ（binary digit の略）。

block（ブロック）　(a)　技術的または論理的理由により，一単位として取り扱われ

る一連の文字または語の集まり。

(b)　ALGOL の用語。プログラムを構成するものであって，宣言の列に文が続いたものを **begin** と **end** でくくったもの。すなわち，任意の宣言及び文をそれぞれ D，S とすると，ブロックは

begin D；…；D；S；…；S **end**

である。ブロックの中の文 S は，それ自身が複合文またはブロックであってもよい。

block data subprogram　specification subprogram と同義。

block error rate（ブロック誤り率）　伝送されたブロックの全数に対する，正しく伝送されなかったブロックの数の割合。

block transfer（ブロック転送）　データのブロックを単一の動作によって転送すること。

Boolean expression（論理式(A)）　論理値が求められる文法単位。ALGOLでは単純論理式と条件節を含む論理式とがある。単純論理式の基本的な構成要素には，論理定数，変数，関数呼出し及び比較式がある。条件節を含む論理式とは，次の形のものである。

"**if** 〈論理式〉 **then** 〈単純論理式〉 **else** 〈論理式〉"

bootstrap（ブートストラップ）　それ自体の働きによって，ある所定の状態に移行するように設計されている手法。例えば，最初の数個の命令によって，それに引き続く全部の命令を入力装置から計算機内に読み込むことができるようにする手法。

borrow（借り）　位取り記数法で表現されている2個の数のあるけたにおける減算の結果が0より小さい場合，1けた上から1を引く操作，またはそのための信号。

BOT　beginning of tape marker の略。

branch instruction　conditional jump instruction と同義。

buffer storage（緩衝記憶装置，バッファ記憶装置）　一つの装置から他の装置へデータを転送する際に，データの流れの速度の相違や事象の発生時点の相違を調整するために用いられる記憶装置。

bus（母線，バス）　多数ある始点の中の任意のものから多数ある終点の中の任意のものにデータを転送するための共通路。

byte（バイト）　一単位として取り扱われるビットの列。多くの場合，8ビットから成る。

[C]

calculator（計算機）　算術演算に特に適し，人手の介入を少なからず必要とする小型のデータ処理装置。

call（呼出し，コール）　指定された閉じたサブルーチンなどに制御を渡すこと。

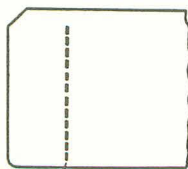
call by name（名前変え(A)）　関数または手続きの仮パラメタが名前変えであるとは，対応する実パラメタをその本体において，そっくりそのまま仮パラメタの位置に埋め込んで実行することをいう。

call by value（値とり(A)）　関数または手続きの仮パラメタが値とりであるとは，対応する実パラメタの値を1回だけ計算して，それを本体において用いることをいう。

calling sequence（呼出し系列）　閉じたサブルーチンを呼び出して制御を渡すために，必要なデータや命令を指定された形式に配列したもの。

card（カード）　一定の形状・寸法を有するカードであって，これに一定の規則に従ってデータを記録できるもの。例えばせん孔カード，エッジカード，OCR 用カード，OMR 用カードなど。

card column（カードの(けた)）　せん孔

カード上で，通常1文字を記録するために割り当てられた一群のせん孔位置。80けたのせん孔カードでは12個の縦に並ぶせん孔位置をいう。

card punch（カードせん孔装置）　計算機などからの指令によって，せん孔カードにデータをせん孔し，記録する装置。

card reader（カード読取装置）　せん孔カード上のデータを読み取る装置。

carry（けた上げ）　位取り記数法で表現されている2個の数のあるけたにおける加算の結果が基数に等しいか，基数を超した場合，1けた上位のけたに1を加える操作，またはそのための信号。

category（項類(C)）　データを数学的なまたは表現上の性質により分類したもの。COBOL では，項類といい，英字，数字，英数字，英数字編集及び数字編集がある。

参考：ALGOL, FORTRAN では type という。

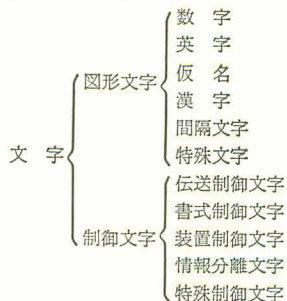
central processing unit（中央処理装置，CPU）　計算機の構成要素の一つであって，命令の解読と実行を制御する回路をもち，通常，演算装置と制御装置とから成る。

central processor　central processing unit と同義。

chained list（連鎖リスト）　リストの一種であって，このリスト中の項目は物理的に続いている必要はなく，各項目は次の項目を示す識別子をもち，これにより順序付けられている。

channel（通信路，チャネル）　信号の一方向の伝送路。

character（文字，キャラクタ）　データの構成，制御，表現に用いられる有限個の要素の集まりの各構成単位であって，一定の約束に基づいて決められているもの。

備考：情報処理の分野において，文字は次のように分類される。

(FORTRAN 用の文字(F))　プログラム言語で許される文字の集まり。下表の文字がある。

A　B…Z
0　1…9
+　−　*　/
△　=　,　.
(　)
‘通貨記号’

注：△は“空白”を示す

character error rate（誤字率）　伝送された文字の全数に対する，正しく伝送されなかった文字の数の割合。

characteristic　exponent と同義。

character recognition（文字認識）　パターン認識の一分野であって，文字の形を自動的手段によって識別すること。

character set（文字の組(C)）　プログラム言語で許される文字の集まり。次頁の表に示す文字がある。

A　B…Z
0　1…9
＋　－　＊　／　＞　＜
△　＝　，　；　．
“(　)
¥

注：△は“空白”を示す

check bit（チェックビット）　文字，ブロックなどの誤りを検査するための付加ビット。例えば奇偶検査ビットなど。

circuit（回路）　往復の両通信路で構成される両方向の伝送路。

circulating storage（循環記憶装置）　一定の時間遅れを作る回路と，増幅整形を行う回路とを組み合せ，データを循環させて記憶させる記憶装置。

class（類（F））　プログラム単位中に現れる英字名が表す文法上の対象を，その現れ方によって第I類から第VIII類の8種類に分類したもの。各類に属する英字名が表す対象は，次のとおりである。

第I類	配列や配列要素
第II類	変　数
第III類	文 関 数
第IV類	組込み関数
第V類	外部関数
第VI類	サブルーチン
第VII類	そのプログラム単位では，サブルーチンとも外部関数とも分類できない外部手続き
第VIII類	ブロック名

clause（句（C））　データ部や環境部の主要な構成単位であって，それぞれデータや機械の特性などを記述する。

clear（クリア）　記憶場所をある定められた状態にすること。通常，ゼロまたは間隔文字に相当する状態にすること。

clock pulse（刻時パルス，クロックパルス）　同期式計算機において，各部の動作の歩調を合わせるために用意された周期的パルス。

closed subroutine（閉じたサブルーチン）　サブルーチンの一種であって，プログラムのある点からこれに移って始められ，終わると元のプログラムの適当な点に制御をもどすように作られたもの。

COBOL（コボル）　事務データ処理を行うためのプログラム言語の一つ。COBOLは，Common Business Oriented Languageの略。

code（コード，符号）（a）　データを表現する方法を規定する明確な約束またはその約束によって表されたもの。

（b）　encodeと同義。

coded decimal notation（符号化10進法）　数の表記法の一つであって，10進法における各けたの10進数字を適当なコードで表すもの。

coder（コーダ）（a）コーディングをする人。

（b）　エンコーダ（encoder）の略。

coding（コーディング）　計算機のコードまたは仮のコードを用いて，プログラムを作る仕事。

collate（照合）　一連のデータの特定項目（一組または数組）における大小関係を順番に判別して処理を行い，該当する項目に付随するデータを一緒に仕分けすること。標準的な処理には，次の二つがある。

（a）　突合せ（matching）　二組のデータの特定の欄どうしを比較し，両者が同じであるかどうかを調べること。

（b）　選別（selection）　指定された条件に従ってデータを選別すること。この条件は，分類または突合せなどの結果として与えられることもある。

collator（照合機）　せん孔カードなどを照合する装置。

COM　computer output microfilmingの略。

combinational circuit（組合せ回路）　任意の時点における出力値が，その時点で

の入力値だけによって決定される論理回路。

備考：組合せ回路は，順序回路の内部記憶能力をもたない場合に相当する。

combinatorial circuit　combinational circuit と同義。

command pulse（指令パルス）　命令の実行に際し，関係各部位の動作を促すために，制御装置から送られるパルス。

comment（注釈（A, F, C））　プログラマの覚え書きなどのための記述。プログラムの実行には何の影響も与えない。ALGOL では，**comment** に続けて注釈を記述し，セミコロンで区切る。または **end** に続けて注釈を記述し，セミコロン，**end** または **else** で区切る。FORTRAN では，注釈行に注釈を記述することができる。注釈行は，行の最初の文字が英字Cである行をいう。COBOL では，NOTE 文または注記項によって注釈を記述することができる。NOTE 文は，NOTE に続けて注釈を記述し，終止符で区切る。注記項は，見出し部に書く。

common block（共通ブロック(F)）　二つ以上のプログラム単位が共有する記憶場所。名前付き共通ブロックと無名共通ブロックとがある。

communication control unit（通信制御装置）　通信回線を介して伝送されるデータ授受に関する制御を行う装置。

comparator（比較器）　二つのデータ項目を比較し，それらが一致したか否かを示す信号を出力する回路。

compiler（コンパイラ）　問題向き言語で書かれたプログラムを計算機向き言語へ翻訳するプログラム。

compiler-directing statement（翻訳指示命令（C））　翻訳指示動詞（COPY, ENTER, NOTE, USE）と，それぞれの作用対象で表現される命令であって，コンパイラに指示した動作をとらせる。

complement（補数）　固定基数表記法において，与えられた数をある特定の数から引いて得られる数であって，基数をbとするとき，与えられた数から次の（a）または（b）によって導かれる数。通常，負の数を表現するのに用いられる。

（a）　bに対する補数（b's complement）　与えられた数の各けたの数をb−1から引き，最下位に1を加え，必要ならばけた上げを行う。

例：2進数00101の2に対する補数は，11011になる。10進数476の10に対する補数は，524になる。

（b）　b−1に対する補数（(b−1)'s complemenet）　与えられた数の各けたの数をb−1から引く。

例：2進数00101の1に対する補数は，11010になる。10進数476の9に対する補数は，523になる。

compound statement（複合文(A)）　プログラムを構成するものであって，文の列を **begin** と **end** でくくったもの。すなわち，任意の文を S とすると，複合文は

begin S;…;S **end**

である。複合文の中の文 S はそれ自身が複合文またはブロックであってもよい。

computer（計算機）　稼動中に操作員が介入することなく，多くの算術演算や論理演算を含むぼう大な計算を行うことのできるデータ処理装置。通常，これは演算装置，制御装置，記憶装置，入力装置及び出力装置の5要素から構成される。

computer oriented language（計算機向き言語）　特定の計算機の構造に拘束されるプログラム言語。アセンブリ言語など。

computer output microfilming（コンピュータ出力マイクロフィルム，COM（コム））　計算機の出力をマイクロフィルム

に記録する方式。

condition（**条件**（**C**））　論理値が求められる文法単位。COBOL では，条件という。条件の基本的な構成要素には，比較条件，字類条件，条件名条件，スイッチ状態条件及び正負条件がある。条件には，論理式の形式でプログラム中に陽に書かれるものと，条件命令の形式によって陰に示されるファイルの終わりなどがある。

conditional jump instruction（**条件付飛越し命令**）　指定した条件が満たされた場合は，次に実行する命令を指定したアドレスから取ることを要求し，満たされない場合は，次に実行する命令を自分の直後のアドレスから取ることを要求する命令。

conditional statement（**条件文**（**A**），**条件命令**（**C**））　それに含まれる論理式または条件の値によって，ある文を実行したり抜かしたりする文法単位。ALGOL では条件文といい，COBOL では条件命令という。FORTRAN には，この用語はない。

console（**制御卓，操作卓**）　制御用キー，スイッチ，指示器などを備え，操作員が必要に応じて計算機の動作に介入し，またはこれを監視するために設けられた卓。保守員のための各種機能を備えたものもある。

constant（**定数**（**F**））　その形がデータの値と特性を明示する文字の列。一般に引用はできるが，変数とは異なり，それに対する値の代入はできない。

contention（**コンテンション**）　一つの通信路上で，二つ以上の端末装置が同時に送信しようとするときに起こる状態。

control break（**制御の切れ目**（**C**））　報告書における出力行の種類と集計は，制御用データ項目と呼ばれるデータ項目の値の変化によって制御される。この変化を制御の切れ目という。

control character（**制御文字**）　定められた特定の文脈中で，ある制御機能を開始したり，変更したり，停止したりするために用いられる文字であって，図形によって表されない。

備考：制御文字は図形文字ではないが，記録を残すなどのために図形で表されることもある。例えば，伝送制御文字などである。

control panel（**制御盤，操作盤**）　制御卓と同じ機能を備えた盤。

control statement（**制御文**（**F**））　プログラムの実行の進行を制御する文であって，GO TO 文，IF 文，CALL 文，RETURN 文，CONTINUE 文，STOP 文，PAUSE 文及び DO 文がある。

control unit（**制御装置**）　計算機の構成要素の一つであって，命令を逐次解読して，計算機内各部に必要な指令を与えることにより，自動的に計算が進行するように制御する装置。

convert（**変換**）　データの意味を完全に保持しながら，表現をある形式から他の形式に変えること。例えば基数変換，コード変換，カードから磁気テープへの変換，AD 変換など。

copy（**転記，コピー**）　ある記憶場所からデータをその原形どおり読み取り，そのデータを同一または異なる物理的形式で他の記憶場所に書き込むこと。例えば磁気テープ上のデータを磁気ディスクに転記するなど。

core storage　magnetic core storage と同義。

counter（**カウンタ，計数器**）　レジスタの一種であって，入力信号を受けることによって内容が1ずつ増加または減少するように構成されたもの。

CPU　Central Processing Unit の略。

cursor（カーソル）　文字表示装置などにおいて，画面上で次に操作される文字位置を表示する印。

cycle time（サイクル時間）　記憶装置中の同一記憶場所に対して，読出し，書込みが始まってから，再び読出し，書込みが行えるようになるまでの最小時間間隔。

cyclic code（巡回符号）　群符号の一種であって，このコードの体系に属するものに，巡回置換を行ったものもまた，この体系に属するようなもの。

cylinder（シリンダ）　同一の回転軸を有する複数枚の磁気ディスクにおいて，回転軸から等距離にあり，アクセスアームを動かさずに書込みまたは読取りができるすべてのトラックの集まり。

［D］

data（データ）　人間または自動的手段によって行われる通信，解析，処理に適するように形式化された事実，概念または指令の表現。

data bank（データバンク）　データのライブラリの集まり。

data base（データベース）　一つ以上のファイルの集まりであって，その内容を高度に構造化することによって，検索や更新の効率化を図ったもの。

data division（データ部（C））　原始プログラムを構成する四つの部の一つであって，3番目に書かれる。実行用プログラムが入力として受け取り，操作し，作り出し，または出力として書き出すデータについて記述する。データ部はファイル節，作業場所節及び報告書節から成る。ファイル節はファイル記述項，分類用ファイル記述項及びレコード記述項から成り，作業場所節はデータ記述項及びレコード記述項から成り，報告書節は報告書記述項及び報告集団記述項から成る。

data item（データ項目（C））　プログラム中の最も基本的な文法単位。名前によって識別され，これに値を与えることができる。変数は文中で値が代入されたり，その値が引用されたりする。COBOLではデータ項目といい，基本的な単位である基本項目と，それをつないだ集団項目とがある。

参考：ALGOL, FORTRAN では variable という。

data link（データリンク）　送信装置から回線を介して受信装置に至る物理的な伝送路と，決められた手順に基づいて設定された論理的な転送経路との総称。

data processing（データ処理）　information processing と同義。

date processing system（データ処理システム）　データ処理を行うために統合された装置，方式，手順及び技術の集まり。

data processor（データ処理装置）　データ処理を行う能力をもつ装置。例えば卓上計算機，せん孔カードシステム(PCS)計算機。

data reduction（データ整理）　データを更に有用な形に変えること。

data signaling rate（データ信号速度）　データ伝送路のデータ伝送能力を，単位時間当たりのデータ伝送量で表すもの。次の式で得られる。

$$\sum_{i=1}^{m}\frac{1}{T_i}\log_2 n_i$$

m：並列通信路の数
T_i：第 i 番目の通信路での最小有意間隔（秒）
n：第 i 番目の通信路における有意状態の数

data transfer rate（データ転送速度）　対応する装置間を通過するデータの単位時間当たりのデータ量。通常，毎秒，毎分，毎時のビット，バイトまたはブロック数で表す。

data transmission（データ伝送）　機械により処理されるべき，または処理されたデータの伝送。

data transmission efficiency（データ伝送効率）　データ伝送路において，単位時間に伝送可能な最大のエレメント，文字，ブロックなどの数に対する，有効に伝送されたものの割合。

debug（デバッグ，（プログラムの）手直し）　プログラムの中の誤りを見つけて直すこと。

decimal digit（10進数字）　10進法の数字であって，0，1，2，3，4，5，6，7，8，9 が用いられる。

decimal notation（10進法）　10を基数とする固定基数表記法。

decimal point　radix point と同義。

decision element　logical element と同義。

decision table（決定表）　問題の解決の手順を記述する方法の一つであって，起こり得る場合をすべて列記し，それに動作を組み合せて表にしたもの。

declaration（宣言（A））　ブロックに使われる量の性質を指定し，それに名前を与える文法単位であって，型の宣言，配列の宣言，スイッチの宣言及び手続きの宣言がある。

decode（復号）　コード化されたデータを元の形に変換すること。

decoder（デコーダ，復号器）　複数個の入力端子と複数個の出力端子とをもつ装置であって，入力端子のある組合せに信号が加えられたとき，その組合せに対応する一つの出力端子に信号が現れるもの。デコーダの作用は，エンコーダ（encoder）の作用の逆に当たる。

defined（確定（F））　変数や配列要素が定義されているとき，その変数や配列要素は確定であるという。

definition（定義（F））　変数や配列要素に値が与えられ，それが引用できる状態になることを，その変数や配列要素が定義されたという。

delay distortion（遅延ひずみ）　所要周波数帯域内で，回線またはシステムの群遅延値が周波数に対して一定でないため起こるひずみ。

delay line storage（遅延線記憶装置）　遅延線を用いた循環記憶装置。例えば，超音波を利用したものなどがある。

designational expression（行き先式（A））　名札が値として求められる式であって，単純行き先式と条件節を含む行き先式とがある。単純行き先式の基本的な構成要素は，名札及びスイッチ呼出しである。条件節を含む行き先式とは，次の形のものである。

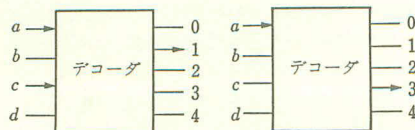
“**if** 〈論理式〉 **then** 〈単純行き先式〉 **else** 〈行き先式〉”

device address（装置アドレス）　入出力装置や補助記憶装置に付けられる装置固有のアドレス。

device control character（装置制御文字）　情報処理システムや通信システムに関連する補助装置を制御するために用いられる制御文字。例えば，装置のオン，オフを制御する文字など。

diagnostic program（診断プログラム）　計算機が正しく働くかどうかを確かめるためのプログラムで，これによって計算機の故障や間違いの状況を知ることができるもの。

dichotomizing search　binary search と

同義。

digital（**ディジタル，計数型**）　データまたは物理量などを数字によって表現することを表す形容詞。

direct access（**直接アクセス**）　データを記憶装置へ書き込んだり，記憶装置から読み出したりする際に，前回書込みまたは読出しが行われた記憶場所とは無関係な記憶場所で書込みまたは読出しが行われるような方法。

direct address（**直接アドレス**）　記憶場所を直接指定するアドレス。

disk pack（**ディスクパック**）　磁気ディスク装置から取り外し可能な複数枚の磁気ディスクで構成される機構。

display unit（**表示装置**）　計算機などからの指令によって，図形や文字などを表示する装置。表示できる内容により図形表示装置，文字表示装置などに分けられる。

division（**部（C）**）　節や段落の集まりであって，見出し部，環境部，データ部及び手続き部がある。

double precision（**2倍精度，倍精度**）　計算機が本来取り扱い得るけた数の，2倍のけた数を取り扱うこと。

dummy argument（**仮引き数(F)**）　手続きの内容を定義するのに用いられる文法単位。例えば，変数や配列など。FORTRAN では，仮引き数といい，手続きが引用されるとき，実引き数と結合される。COBOL には，このような概念はない。

dummy statement（**空文(A)**）　実行時には何もしない文であって，飛越し文の行き先の名札を示すときなどに用いる。

duplicate（**複製**）　ある記憶場所からデータをその原形どおり読み取り，そのデータを同一の物理的形式で他の記憶場所に書き込むこと。例えば元のせん孔カードと同じせん孔形式で，新しいせん孔カードを複製するなど。

［E］

echo checking system（**エコーチェック方式**）　loop checking system と同義。

edge punched card（**エッジカード**）　カードの側端にせん孔機によってデータのせん孔を行い，紙テープ読取り装置で読み取ることができるようにしたもの。

edit（**編集**）　後の操作のためにデータを整えること。例えば印刷のためにデータの配置替え，データの追加，不必要なデータの削除，書式制御，コード変換，ゼロ抑制などをすること。

EDP　electronic data processing の略。

electronic data processing（**電子データ処理，EDP（イーデーピー）**）　主として電子的手段によって行われるデータ処理。

element（**エレメント**）　(a)　文字が形づくられる等長多単位符号の各素子。

(b)　要素，素子。

(c)　ソフトウェアを構成するプログラムの部分。

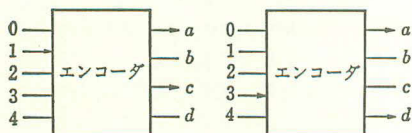
element error rate（**エレメント誤り率**）　伝送されたエレメントの全数に対する，正しく伝送されなかったエレメントの数の割合。

emulation（**エミュレーション**）　ある計算機が別種の計算機用に書かれたプログラムを，特別な機構及びプログラミング技法を使用して，そのまま実行できるようにすること。

emulator（**エミュレータ**）　エミュレーションを行う装置またはプログラム。

encode（**コード化，符号化**）　コードを用いて，通常，元の形へ変換できるような仕方によってデータを変換すること。

encoder（**エンコーダ，符号器**）　複数個の入力端子と複数個の出力端子とをもつ装置であって，ある1個の入力端子に信号が加えられたとき，その入力端子に対応

する出力端子の組合せに信号が現れるもの。コーダともいう。

end-arround carry（循環けた上げ） 演算の結果，最上位に生じたけた上げを，最下位へ回して加えること。

END line（END 行（F）） プログラム単位の最後の行であって，END と書く。

end of tape marker（テープ終端マーカ） 磁気テープ上の記録可能な範囲の終端を示す物理的な印。例えばアルミはくの反射マーカなどがある。

entry（記述項（C）） 句の続いた組であって，終止符と空白とで止める。原始プログラムの見出し部，環境部，データ部に書く。

environment division（環境部（C）） 原始プログラムを構成する四つの部の一つであって，2番目に書かれる。内容は，翻訳用計算機と実行用計算機の特性，使用ファイル，外部媒体，アクセス技法などである。環境部は，構成節と入出力節とから成る。構成節は，翻訳用計算機段落，実行用計算機段落及び特殊名段落から成り，入出力節は，ファイル管理段落及び入出力管理段落から成る。

EOT end of tape marker の略。

error burst（バースト誤り） 二つの隣接した誤りビットの間の正しいビットの数が，ある与えられた数 x よりすべて小さいような誤りビットの一群。

備考：バーストの最後の誤りビットと，次のバーストの最初の誤りビットとの間は，x 以上離れている。バースト誤りを記述するときは，数 x を明記しておかなければならない。

error control system（誤り制御方式）

(a) 計算機，通信回線などに発生したデータの誤りの有無を検出し（誤り検出方式），または更に訂正する方式（誤り訂正方式）の総称。

(b) データ伝送において，受信データの誤りの有無を検出し，または更に訂正する方式の総称。

備考：誤りの検出と訂正の代表的な方式は，次のとおりである。

- 誤りの検出
  - 自動連送方式
  - 返送照合方式
  - 誤り検出符号方式
  - 妨害検出方式
- 誤りの訂正
  - 再送訂正方式
  - 自己訂正方式

error correcting code（誤り訂正符号） コードを構成するビットの中に誤りがあったとき，それを訂正できるような規則で構成された冗長符号。

error correcting system（誤り訂正方式） データの誤りの検出だけでなく，その訂正も行う誤り制御方式。誤り訂正符号を用いた誤り制御方式を特に指す場合もある。

error detecting code（誤り検出符号） コードを構成するビットの中に誤りがあったとき，それを検出できるような規則で構成された冗長符号。

error detecting code system（誤り検出符号方式） データのブロックに検査ビットを付加して誤りを検出する方式。検査ビットとしては，例えば，水平方向で"1"の個数を2進法で表し，この一部または全部を使用することが多い。

error detecting system（誤り検出方式） データの誤りの有無の検出だけを行う誤り制御方式。

exclusive OR（排他的論理和） P 及び Q

を二つの論理変数とするとき，次の表によって定まる論理関数 $P \oplus Q$ を“P と Q との排他的論理和”という。

表

P	Q	$P \oplus Q$
0	0	0
0	1	1
1	0	1
1	1	0

executable program（実行可能プログラム（F）） 計算機が受け入れ得る形式になっている一連の命令または文。FORTRANでは，一つの主プログラムまたは一つの主プログラムと幾つかの副プログラムや外部手続きとから成る。

executable statement（実行文（F）） 実行時の動作を指定する文であって，代入文，制御文及び入出力文がある。

execution cycle（命令実行段階） 制御装置が新しい命令を取り出し終わってから，その実行が終わるまでの動作段階。

exponent（指数） 浮動小数点表示における指数部の数。

expression（式（A, F）） 基本的な構成要素と演算子を組み合せた文法単位。これを評価することによって，一つの値が求められる。ALGOLでは，算術式，比較式，論理式及び行き先式がある。FORTRANでは，算術式，関係式及び論理式がある。

external storage（外部記憶装置） 中央処理装置が，入出力チャネルを通してデータを書き込んだり読み出したりできる記憶装置。

extract（抽出） (a) 指定された語の指定された（一つまたはそれ以上の）位置にあるビット，バイトまたは文字を取り出すこと。

(b) 項目の集まりから指定された条件を満たす項目を選び出すこと。

［F］

fetch cycle（命令取出段階） 制御装置が前の命令の実行中または実行終了後に，次に実行すべき命令を記憶装置から取り出し始めてから取り出し終わるまでの動作段階。

field（欄，フィールド） レコード内の一つの項目のための特定な領域。

file（ファイル） 情報処理の目的で，一単位として取り扱われる関連したレコードの集まり。

（ファイル（F, C）） レコードの集まり。FORTRANでは，記録を書き出した順番にしか読み込めない。COBOLでは，レコードを書き出した順番にも，指定したレコードをランダムにも読み込める。

file layout（ファイル様式，ファイルレイアウト） ファイル中のデータや語の配置及び構造のことで，その構成要素の順序と大きさの仕様を含む。

file maintenance（ファイル保守，ファイル維持） データを追加，変更または削除することにより，ファイルを最新の状態に保つこと。

file protection ring（ファイル保護リング） write enable ring と同義。

first-in first-out（先入れ先出し方式） 最も古い項目が先に取り出されるように管理する方式。

first level definition（第1階の定義（F）） 数値に関する定義において，変数や配列要素の値が，数値そのものとして引用される場合の定義をいう。

fixed-point part mantissa（仮数） 浮動小数点表示における仮数部の数。

fixed-point representation（固定小数点表示） 小数点が数字の並びの一端を基準にして，決まった位置に固定されてい

るような基数表記法。

fixed radix notation（固定基数表記法） 各けたが，すべて同じ基数をもつような基数表記法。

fixed storage（固定記憶装置） 自動的に書込みができない記憶装置であって，読出し専用に使われるもの。通常，定数，常用ルーチンなどを入れて用いる。必要に応じて手動的に書き込むことが可能になっているものもある。

fixed word length（固定語長） 記憶装置の基本語長が一定である形式。

flag（標識，マーク） (a) しるしを付けるために用いられるビット，バイト，文字または語。

(b) 欄，語，データの項目，またはファイル，レコード，ブロックなどのデータの集まりの始まりまたは終わりを示す記号。

flip-flop（フリップフロップ） “双安定回路” bistable circuit と同義。

備考：単安定回路としてもまれに用いられることがある。

floating head flying head と同義。

floating-point representation（浮動小数点表示） 数を指数部と仮数部によって表す表記法であって，ある基数を指数部の数で示した回数だけ累乗して得られる数値に仮数部の数を掛けたものが，その数値となる。

例：0.0001234（すなわち 0.1234×10^{-3}）を，浮動小数点表示では，通常，次のように表す。

.1234　E-03
仮数部　指数部

flowchart（流れ図） 動作，データ，それらの流れ，使用する装置などを記号を用いて表すことによって，問題の定義，分析または解法を図式的に表現したもの。

flying head（浮動ヘッド） 磁気ディスク装置や磁気ドラム装置において，ディスクやドラムの回転によって生じる気流または外部からの加圧気体によって，回転面上に浮動させて用いる磁気ヘッド。

formal parameter（仮パラメータ(A)） 手続きの内容を定義するのに用いられる文法単位。ALGOLでは，仮パラメタといい，手続きが呼ばれるとき，実パラメタとの対応がとられる。

format（書式(A, F)） データの形及びこの形を指定する文字の列。ALGOLでは，書式といい，入出力のための外部媒体上のデータに対して用いる。FORTRANでは，書式といい，入出力のための外部媒体上のデータに対して用いる。

format effector（書式制御文字） 入出力媒体における情報の配列や位置を制御するために用いられる制御文字。

for statement（繰返し文(A)） 繰返し節に続けて一つの文を書いたもの。繰返し節とは，**for** と **do** の間に繰返しに関する条件を記述した文法単位である。繰返し文は，その中に含まれる文を，条件が満たされている間繰り返し実行する。

FORTRAN（フォートラン） 本来，数値計算を行うためのプログラム言語の一つ。FORTRANは，Formula Translator の略。

forward read（順読み） 紙テープ読取り装置や磁気テープ装置などで，順方向に媒体を進めながら，記録されているデータを読み取ること。

full adder（全加算器） adder (b) と同義。

full duplex（全二重） データを回線上のどちらの方向にも伝送することができ，かつ，両方向同時に伝送することができる方式。

function（関数(A, F)） 式の中で用いることによって，一つの値が定まる手続

き。関数の呼出し及び関数の引用は，関数名の後ろに引き数の並びを添える。

function subprogram（関数副プログラム(F)） 副プログラムの一つであって，FUNCTION文を先頭とする幾つかの文によって定義される外部関数。式の中で引用され，算術的な値や論理値を与える。

［G］

gate（ゲート） (a) 1個以上の入力端子と1個の出力端子とをもち，その出力が入力の状態によってだけ定まる回路。

(b) 集積回路（IC）の中の回路数の単位。

general purpose computer（はん(汎)用計算機） 広い範囲の問題を処理できるように設計されている計算機。

general purpose register（はん(汎)用レジスタ） 計算機の中央処理装置の中にあって，累算器，指標レジスタなど多目的に使用されるレジスタ。通常，複数個用意されており，アドレスで指定ができる。

generator（ゼネレータ，生成プログラム） パラメタを受け取って，ある骨組に従った機械語のプログラムを作り出すプログラム。

go to statement（飛越し文(A)） “go to 〈行き先式〉”の形で表現される文であって，文が書いてある順序で決められた通常の実行順序をそこで中断し，行き先式の値が示す名札をもつ文へ実行を移す働きをする。

graphic character（図形文字） 特定の図形をもつ文字であって，表音または表意などに用いられるもの。

Gray code（グレイコード） reflected binary code と同義。

group code（群符号） コードの集まりが数学上の群を作るような冗長符号。

［H］

half adder（半加算器） 2個の入力端子と2個の出力端子とをもち，出力信号が入力信号に対し，次の表の関係にある回路。

この回路を2個用いて，2進加算器の1けた分を構成できるので半加算器という。

表

入力		出力	
X	Y	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

half duplex（半二重） データを回線上のどちらの方向にも伝送することができるが，両方向同時には伝送することができない方式。

hamming distance（ハミング距離） 同じ語長をもつ二つの2進コードに対応する各けたを比較して，異なっているけたの個数。例えば，01011100 と 11011111 のハミング距離は3である。

hard copy（ハードコピー） 機械の出力を目で見て読める形で記録したもの。

hardware（ハードウェア） データ処理に用いられる物理的な機器。ソフトウェア（software）の対照語。

hardware representation（金物表現） 基準言語を，特定の計算機の文字の組により，一定の規則に従って変換し，そのコンパイラによって読めるようにしたもの。

head（ヘッド） 記録媒体上にデータを書き込んだり，書かれたデータを読み取ったり消去したりするための機構。例えば磁気ヘッド，光電式ヘッド及びせん孔ヘッドなど。

hexadecimal digit（16進数字）　16進法の数字であって，10進数字の外に，10進法の10から15までを表す特別な文字（通常は，英字のうちの6個，例えば，A，B，C，D，E，Fで代用する）が用いられる。

hexadecimal notatiom（16進法）　16を基数とする固定基数表記法。

high level language（高水準言語）　特定の計算機の構造に拘束されることの少ないプログラム言語。

hopper（ホッパ）　カードなどを扱う装置において，送り機構に順序よく送り出せるようにカードなどを積んでおく部分。スタッカと対比される。

horizontal check（水平検査）　媒体に記録された2進コードの検査の方式であって，媒体の運動方向に対し，平行な方向のビットについて奇偶検査などを行うこと。

hybrid computer（ハイブリッド計算機）データにアナログとディジタルの両方の表現を用いる計算機。

［I］

identification division（見出し部(C)）原始プログラムを構成する四つの部の一つであって，最初に書かれる。内容は，プログラム名，プログラムを書いた日付け，プログラムを翻訳した日付けなどである。見出し部は，プログラム名段落などから成る。

identifier（識別子）　データの項目を表示したり名付けたりするための一連の文字の列。名前（A），一意名（C），変数，配列，手続きなどを識別するのに用いる文字の列。

identify（識別する(F)）　着目している対象が，それそのものであることを認識すること。FORTRANでは，識別するということばを，引用するという意味と，名前が呼ばれるという意味で用いる。

imperative statement（無条件命令（C））論理式または条件に制御されることなく，定められた動作を行う文法単位。
参考：ALGOL　では，unconditional statement という。

index（索引）　ファイルまたは文書に関する見出しが順序よく記入されている表であって，その内容または在り場所などの表示がなされているもの。

index register（指標レジスタ）　命令を実行する直前に，そのアドレスを変更する機能をもった計算機において，それに用いる変更子を保持しているレジスタ。

indirect address(間接アドレス）　直接アドレスまたは別の間接アドレスを記憶している場所を指定するアドレス。

information（情報）　データを表現するために用いた約束に基づいて人間がデータに割り当てた意味。

information bit（情報ビット）　情報を表現するビット。

information interchange（情報交換）異なるシステム間でも相互に情報が利用できるように，一つのシステムから他のシステムへ情報を伝えること。

information processing（情報処理）　必要な情報を得るために，データに対して行う一連の作業。

information separator（情報分離文字）論理的に情報を分離し，区別するために用いられる制御文字。通常，分離される情報の大小関係に従って，ファイル分離文字，グループ分離文字，レコード分離文字，ユニット分離文字の4種類が使い分けられる。
備考：各情報分離文字の名称と，分離される情報の単位とは，必ずしも一致しない。

inhibit（抑止）　P及びQを二つの論理変数とするとき，次の表によって定まる論理関数Rを“PをQで抑止する”という。

表

P	Q	R
0	0	0
0	1	0
1	0	1
1	1	0

inhibit circuit（抑止回路）　1個以上の入力端子，1個以上の制御端子及び1個の出力端子をもち，制御信号が特定の条件を満たした場合には，入力が“1”であっても“0”であっても出力端子に“0”を出力する回路。

input device　input unit と同義。

input-output channel（入出力チャネル）　中央処理装置の指令に基づいて，入出力装置と記憶装置との間で，中央処理装置の動作と独立に，データの授受を行うための装置。

input-output control unit（入出力制御装置）　入出力を制御する装置の総称。

input-output device　input-output unit と同義。

input-output statement（入出力文（F））　外部媒体との間の入出力や関連動作を行う文であって，READ 文，WRITE 文及び補助入出力文（REWIND 文，BACKSPACE 文及び ENDFILE 文）がある。

input-output unit（入出力装置）　入力装置及び出力装置並びに入力及び出力の両機能を兼ね備えた装置の総称。

input unit（入力装置）　計算機の構成要素の一つであって，計算機外からデータを読み込む装置。

instruction（命令）　機械語の一単位であって，計算機に演算その他の一定の動作を指示するもの。命令は，通常，命令コードと一つ以上のアドレスから構成される。アドレスには，命令の記憶場所を示すもの（命令アドレス）とオペランドの記憶場所を示すもの（データアドレス）とがある。

instruction address register（命令アドレスレジスタ）　制御装置の一部であって，逐次制御を行うために，次に読み出すべき命令のアドレスを保持するレジスタ。

instruction register（命令レジスタ）　制御装置の一部であって，記憶装置から読み出された命令を受け取り，それを実行するために一時保持しておくレジスタ。

integrated data processing（集中データ処理）　種々のデータを1箇所に集め，ある目的に従って，それぞれに必要な処理（計算）をし，必要ならば，結果を返送すること，またはそのような方式。

interface（インタフェース）　二つ以上の構成要素の境界または境界において共用される部分。二つの装置を連結するハードウェア構成要素である場合，二つ以上のプログラムによって共用される記憶装置の一部またはレジスタである場合などがある。

interference detection system（妨害検出方式）　通信路で生じた瞬断，ピーク雑音など誤りの原因となるような妨害を検出し，誤りを検出する方式。

internal storage（内部記憶装置）　中央処理装置が，直接指定してデータを書き込んだり読み出したりできる記憶装置。

interpreter（インタプリタ，通訳プログラム）　原始言語の命令を一つごとに翻訳・実行するプログラム。

interruption（割込み）　一つのプログラムの実行をハードウェア的手段で中断し，

後で再開できるようにして他のプログラムの実行に移ること。

intrinsic function（組込み関数(F)）　プログラマが定義しないで使うことのできる関数。例えば絶対値や剰余を計算する関数など。

inverted AND　NAND と同義。

inverted OR　NOR と同義。

item（項目）　(a)　データの集合の要素。例：ファイルはレコードのような幾つかの項目から成り，更にそのレコードは他の項目から成っている。

(b)　情報処理の目的で，一単位として取り扱われる一連の文字または語の集合をいうこともある。

［J］

jump instruction（飛越し命令）　条件付きまたは無条件に，次に実行する命令を，指定したアドレスから取る（指定したアドレスにある命令へ制御を渡す）ことを要求する命令。

［K］

kana（仮名）　日本語を書き表すときに用いられる表音図形文字。片仮名と平仮名の２種類がある。

kanji（漢字）　日本語を書き表すときに用いられる表意図形文字。

key（見出し，キー）　データの集まりに含まれるかまたは付加される文字列であって，その集まりを識別したり，制御したりする情報をもつもの。

keyboard（けん盤，キーボード）　キーを一定の規則に従って配列した盤であって，そのキーを押すとそれに対応した信号を送り出すもの。

keypunch（せん孔機）　けん盤を手動で操作することによって，せん孔カードなどにデータをせん孔し，記録する装置のこと。

［L］

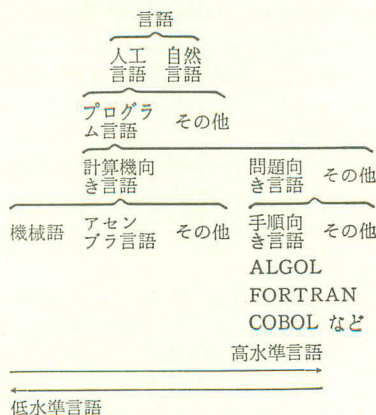
label（ラベル）　(a)　データの項目を指すために用いられる一連の文字の列。

(b)　ボリュームやファイルの始まりまたは終わりにあって，そのボリュームやファイルを識別したり，境界を定めたりするために用いられるブロック。

（名札(A)）　文の先頭につけて文の引用や飛越しの行き先を識別する文字の列。

参考：FORTRAN では statement label, COBOL では Procedure name という。

language（言語）　情報を伝達するために使われる表現，約束及び規則の集まり。

参考：情報処理の立場から，言語は次のように分類されている。

language processor（言語プロセッサ）　あるプログラム言語を処理するために必要な翻訳などの機能を果たすプログラム。

language translator（言語翻訳プログラム）　ある言語を他の言語へ，特にあるプログラム言語を他のプログラム言語へ翻訳するプログラム。

last-in first-out（後入れ先出し方式）　最も新しく加えられた項目が先に取り出されるように管理する方式。

latency（待ち時間）　制御装置が記憶装置からのまたは記憶装置へのデータの転送を要求してから，転送が実際に開始されるまでの時間。

least significant（最下位）　位取り記数法において，最も重みの小さいけた位置。

letter（文字）　A, B, …, Z 及び a, b, …, z の 52 個の図形文字。

library of data（データのライブラリ）　一つの適用業務に関係したファイルの集まり。例えば在庫管理において在庫品管理のファイルの集まりが，データのライブラリを形成する。

library of program（プログラムライブラリ）　標準化されたまたは検査済みの，プログラム及びサブルーチンの集まり。

light pen（ライトペン）　先端に光電素子を備えたペン状の器具であって，表示装置の画面上の光点を検出し，計算機に入力するために用いられるもの。

line（行(F, C)）　プログラムの書き方における単位。定まった個数の文字の列から成る。FORTRAN では，開始行，継続行，注釈行及び END があり，文は一つの開始行だけかまたはそれに続く幾つかの継続行から成る。COBOL では，前の行，後の行及び空白行がある。一つの語や文字列が2行にわたるとき，続きの行を後の行といい，このようにして続いていない行を前の行という。

line printer（ラインプリンタ，行印字装置）　各けたごとに独立した印字機構があり，横書きに行を一度に印字できる印字装置。

link（連結，リンク）　(a)　二つのプログラムを結び合わすこと，またはその手法。

(b)　呼び出されたプログラムから元のプログラムへもどることを可能にするための手段。

linkage（リンケージ）　link と同義。

list（リスト，並び）　項目の順序付けられた集まり。

list processing（リスト処理）　リスト形式のデータを処理すること。項目の物理的位置を変更することなく，論理的順序を変更できるようにするには，通常，連鎖リストが使われる。

literal（直定数(C)）　その形がデータの値と特性を明示する文字の列。一般に引用はできるが，変数とは異なり，それに対する値の代入はできない。

参考：FORTRAN では constant という。

load point（ロードポイント）　磁気テープ上の記録開始可能な位置。

local（局所的(A)）　あるブロックに関して名前が局所的であるとは，使われている量の名前がそのブロックで宣言されている状態をいい，その名前は，このブロックの外では参照できない。

logical circuit（論理回路）　論理素子を用いて論理関数を構成する回路。

logical element（論理素子）　計算機などの回路で論理演算を行う回路の最小構成要素。論理素子の作用は，“論理和”，“論理積”，“否定” などの論理演算で表現できる。

logical expression（論理式(F)）　FORTRAN の論理式の基本的な構成要素には，論理定数，論理型の変数名，配列要素名，関数の引用及び関係式がある。

logical operation（論理演算）　四則演算以外の演算。1ビットごとの論理和，1ビットごとの論理積，比較，抽出など。

logical operator（論理作用素(A)，論理演算子(F, C)）　論理式または条件を構成

する要素。これらは，次の表のとおりである。

表

演算	ALGOL	FORTRAN	COBOL
	論理作用素	論理演算子	論理演算子
否定	¬	NOT	NOT
論理積	∧	AND	AND
論理和	∨	OR	OR
含意	⊃		
同値	≡		

logic circuit　logical circuit と同義。

logic design(論理設計)　方式設計と実装設計との中間段階であって，主として記号化された論理素子の組合せによって計算機の設計をすること。広い意味では，方式設計を含む場合もある。

longtitudinal check　horizontal check と同義。

loop(ループ)　プログラムの中で繰り返して実行できるようになっている一群の命令。

loop checking system (返送照合方式)　送られたデータを全部送信側に返送し，元のデータと照合することにより，誤りを検出する方式。echo checking system ともいう。

[M]

machine address (機械語アドレス)　機械内部でプログラムを実行する際に，直接使用される固有のアドレス。

machine language(機械語)　機械が直接解読し，実行することができる言語。

macroinstruction (マクロ命令)　同一の原始言語で記述された一連の命令文によって置き換えることができる原始言語の中の命令。マクロ命令中にパラメタを指定できる場合には，その値が置き換えられる命令文の中にそう入される。

magnetic core storage (磁心記憶装置)　磁心の残留磁束の向きによってデータを蓄える記憶装置。通常，磁心マトリックスが用いられ，選択された縦横の線の交差点にある磁心にデータが書き込まれたり，そこから読み出されたりする。

magnetic disk(磁気ディスク)　磁性材料で被膜された平らな円板であって，一定の磁気記録方式により，データを記録することができるもの。

magnetic disk handler　magnetic disk unit と同義。

magnetic disk unit (磁気ディスク装置)　計算機などからの指令によって，磁気ディスク上にデータを記録し，また，磁気ディスク上に記録されているデータを読み取る装置。磁気ディスクを回転させるための駆動機構，磁気ヘッド及びそれに付随する制御機構を含む。

magnetic drum (磁気ドラム)　磁性材料で被膜された円筒であって，一定の磁気記録方式により，データを記録することができるもの。

magnetic drum unit (磁気ドラム装置)　計算機などからの指令によって，磁気ドラム上にデータを記録し，また，磁気ドラム上に記録されているデータを読み取る装置。磁気ドラムを回転させるための駆動機構，磁気ヘッド及びそれに付随する制御機構を含む。

magnetic head (磁気ヘッド)　磁気ドラム，磁気ディスク，磁気テープなどの磁性面にデータを書き込んだり，書かれたデータを読み取ったり消去したりするための機構。

magnetic ink character inscribe (磁気インク文字記録機)　けん盤を手動で操作することによって，用紙上に磁気インキ文字を記録する装置。

**magnetic ink character reader(磁気イ

ンク文字読取装置，MICR） 磁化されやすい性質の特殊なインキで書かれた文字を読み取る装置。

magnetic tape（磁気テープ） 磁性材料で被膜されたテープであって，一定の磁気記録方式により，データを記録することができるもの。

magnetic tape cassette（磁気カセットテープ） 所定の形状の小箱に組み込んだ磁気テープ。

magnetic tape cassette handler magnetic tape cassette unit と同義。

magnetic tape cassette unit（磁気カセットテープ装置） 計算機などからの指令によって，磁気カセットテープ上にデータを記録し，また，磁気カセットテープ上に記録されているデータを読み取る装置。

magnetic tape handler magnetic tape unit と同義。

magnetic tape unit（磁気テープ装置） 計算機などからの指令によって，磁気テープ上にデータを記録し，また，磁気テープ上に記録されているデータを読み取る装置。磁気テープの駆動機構，磁気ヘッド及びそれに付随する制御機構を含む。

magnetic thin film storage（磁気薄膜記憶装置） 強磁性金属薄膜を用いて，一定の磁気記憶方式によりデータを記憶する装置。

main program（主プログラム） プログラムの骨幹になる部分。プログラムは，主プログラムと閉じたサブルーチンとから構成されるのが普通である。main routine, master routiue ともいう。

（主プログラム（F）） 実行可能プログラムを構成する主要な文法単位であって，少なくとも一つの実行文を含み，副プログラム文以外の文や注釈行から成り，END 行で終わる。最初に実行され，かつ，他の副プログラムや外部手続きから引用されない。

main routine（主ルーチン） main program 主プログラムと同義。

main storage（主記憶装置） 内部記憶装置であって，プログラムにより直接，アドレスを指定できるもの。

marginal check（限界検査） 計算機などにおいて，ある動作条件（例えば供給電圧）の許容範囲内で要求されている機能を満足しない部分を見付け出すために，その動作条件を定格値から変化させて行う検査。

mark（マーク） flag と同義。

mark reader（マーク読取装置） 特定のカードまたは用紙に付けられた印を光学的または電磁気的に読み取る装置。

mask（マスク） (a) ビット，バイトまたは文字の並びからその一部を抽出したり削除したりするために，他のビット，バイトまたは文字の並びを用いること。
(b) 上の目的に用いられるビット，バイトまたは文字の並び。

master file（マスタファイル，基本フィル） ある仕事で基本となって使用されるファイル。このファイルは内容を更新されながら，継続して用いられる。

master routine main program 主プログラムと同義。

matrix（マトリックス） 同じ部品を多数縦横に配列し，それらを網状に導線によって連結して構成された装置。ダイオードマトリックス，磁心マトリックスなどがあり，前者はエンコーダ，またデコーダとして，後者は記憶装置として使用される。

memory（メモリ） storage と同義。

merge（併合） 同じ規則によって項目が並べられている二つ以上の集まりを，その

規則による一つの集まりにすること。

message（メッセージ） 初めと終わりが明確に規定されたデータであって，情報伝達を目的としたもの。

MICR magnetic ink character reader の略。

microinstruction（マイクロ命令） 命令を更に分解して，ハードウェアに基本動作を指示するもの。

microprogram（マイクロプログラム） 一連のマイクロ命令であって，専用の記憶装置または記憶域に記憶されるもの。

mixed radix notation（混合基数表記法） 各けたが，必ずしも同じ基数をもたないような基数表記法。

modem（モデム） modulator-demodulator の略。

modifier（変更子） アドレスを変更するために用いる量。

modifier register index register と同義。

modulation rate（変調速度） 秒を単位として測定した変調の最小有意間隔の逆数値。

modulator-demodulator（変復調装置） 通信路を通して伝送される信号の変調及び復調を行う装置。modem ともいう。

monostable circuit（モノステーブル回路） 一つの安定状態と一つの不安定状態とをもつトリガ回路，トリガ入力の印加により不安定状態になり，回路で決まる一定時間後の安定状態にもどる。

monostable trigger circuit monostable circuit と同義。

most significant（最上位） 位取り記数法において，最も重みの大きいけた位置。

move transfer と同義。

multiplexor channel（多重チャネル，マルチプレクサチャネル） 複数台の入出力装置を接続し，それらを同時に動作させることのできる入出力チャネル。データ転送は，バイト単位またはブロック単位で行われる。

multipoint system（分岐方式，マルチポイント方式） 回線の両端に接続される端末装置など以外に，その間に他の端末装置などが接続されている方式。

multiprocessing（多重プロセッシング，マルチプロセッシング） 複数個の処理装置をもつ一つのデータ処理システムにおいて，幾つかの仕事を分担して並列に実行すること。

multiprocessor（多重プロセッサ，マルチプロセッサ） 主記憶装置を共用し，かつ同時に動作可能な複数個の処理装置をもつ計算機またはデータ処理システム。

multiprogramming（多重プログラミング，マルチプログラミング） 一つの処理装置で複数個のプログラムを，割込みなどの手法により，見掛け上同時に実行すること。

［N］

named（名前が呼ばれる(F)） 変数や配列要素が，プログラム中で識別はされるが，その値は必ずしも使用可能でない場合，その変数や配列要素は名前が呼ばれるという。代入文の左辺の変数や READ 文の入力並び中の変数などは，この場合に当たる。

NAND（否定積） P 及び Q を二つの論理変数とするとき，次の表によって定まる論理関数 $\overline{P \cdot Q}$ を "P と Q との否定積"

表

P	Q	$\overline{P \cdot Q}$
0	0	1
0	1	1
1	0	1
1	1	0

という。

NAND circuit（否定積回路，NAND 回路）　2個以上の入力端子と1個の出力端子とをもち，すべての入力端子に“1”が入力された場合にだけ，出力端子に“0”を出力する回路。

NDRO storage　nondestructive readout storage の略。

nondestructive readout storage（非破壊読出し記憶装置）　読出し操作によって，記憶されているデータが消えず，したがって，データを保持するために再び書込みをする必要のない記憶装置。例えば磁気ドラム，磁気薄膜などを用いた記憶装置。NDRO storage ともいう。

non-executable statement（非実行文(F)）　プログラムを実行するための準備に関する情報を指定する文であって，宣言文，DATA 文，FORMAT 文，文関数定義文及び副プログラム文がある。

non-impact printer（非衝撃式印字装置，ノンインパクトプリンタ）　印字に際し，機械的衝撃を用いない印字装置。感熱式印字装置，静電式印字装置，写真式印字装置など。

nonlocal（非局所的(A)）　あるブロックに関して名前が非局所的であるとは，使われている量の名前が，そのブロックで宣言されていない状態をいい，その名前は，このブロックの内でも外でも同じ対象を表す。

nonvolatile storage（持久記憶装置）　蓄えられたデータを保持するのにエネルギーを必要としない記憶装置。電源が切れても蓄えられたデータが消滅しないことが特徴である。磁心，磁気ドラム，磁気ディスクなどを用いた記憶装置。

NOR（否定和）　P 及び Q を二つの論理変数とするとき，次の表によって定まる論理関数 $\overline{P+Q}$ を“P と Q との否定和”という。

表

P	Q	$\overline{P+Q}$
0	0	1
0	1	0
1	0	0
1	1	0

NOR circuit（否定和回路，NOR 回路）　2個以上の入力端子と1個の出力端子とをもち，すべての入力端子に“0”が入力された場合にだけ，出力端子に“1”を出力する回路。

normalization（正規化）　(a)　ある量をあらかじめ定められた範囲内に収まるように調整すること。

(b)　浮動小数点表示において，同じ値を表す指数部と仮数部との組合せが二つ以上あるとき，その中の一つを標準の表現と定め，値を変えることなしに他の表現の数を標準の表現に変換すること。

例：.1234E-03　　標準の表現
.01234E-02
.00001234E+2　｝標準以外の表現

NOT（否定）　P をある論理変数とするとき，次の表によって定まる論理関数 $\bar{P}$ を“P の否定”という。

表

P	$\bar{P}$
1	0
0	1

$\bar{P}$ を P', $\sim P$, $\neg P$ などと書くこともある。

notation（表記法）　データを表示するための文字の集まりとその使い方の規則。

NOT circuit（否定回路，NOT 回路）　1個の入力端子と1個の出力端子とをもち，入力端子に“0”が入力された場合にだけ，出力端子に“1”を出力する回

路。

numerical character（数字）　0, 1, …, 9 の数を表す 10 個の図形文字。

［O］

object program　target program と同義。

OCR　optical character reader の略。

octal digit（8 進数字）　8 進法の数字であって，0, 1, 2, 3, 4, 5, 6, 7 が用いられる。

octal notation（8 進法）　8 を基数とする固定基数表記法。

off line（オフライン，非直結）（a）データの転送過程において，人手の介入を必要とする状態。

（b）中央処理装置の直接制御下にない状態。

one address instruction（1 アドレス命令）　命令の形式の一種であって，アドレスを 1 個含むもの。

one plus one address instruction（1+1 アドレス命令）命令の形式の一種であって，2 個のアドレスを含み，その 1 個はオペランドのアドレスであり，他の 1 個は次に実行すべき命令のアドレスであるもの。

on line（オンライン，直結）（a）データの転送過程において，人手の介入を必要としない状態。

（b）中央処理装置の直接制御下にある状態。

open subroutine（開いたサブルーチン）サブルーチンの一種であって，プログラムの一部として命令の系列の中に直接組み込まれるもの。

operand（オペランド，演算数）　演算の対象となるもの。

例：$A+B$ という演算では，A は第 1 オペランド，B は第 2 オペランドと呼ばれる。

operating system（オペレーティングシステム）　プログラムの実行を制御し，スケジューリング，入出力制御，記憶域割当て，データ管理，コンパイル，デバッグ，課金処理及び関連する諸サービスを提供するソフトウェア。

operation code（命令コード）　命令の中にあって，演算その他の動作を指示するコード。

operation time（演算時間）　ある命令（加算，乗算，除算など）について，命令実行段階に費やされる時間に命令取出し段階の時間を加えたもの。ただし，前者だけを指す場合もある。

optical character reader（光学式文字読取装置）　機械によって印字された文字または手書きの文字を，光学的手段により読み取る装置。OCR ともいう。

OR（論理和）　P 及び Q を二つの論理変数とするとき，次の表によって定まる論理関数 $P|Q$ を “P と Q との論理和” という。$P+Q$ を $P \vee Q$ などと書くこともある。

表

P	Q	$P\|Q$
0	0	0
0	1	1
1	0	1
1	1	1

OR circuit（論理和回路，OR 回路）　2 個以上の入力端子と 1 個の出力端子とをもち，少なくとも 1 個の入力端子に “1” が入力された場合にだけ，出力端子に “1” を出力する回路。

output device　output unit と同義。

output unit（出力装置）　計算機の構成要素の一つであって，計算機外へデータを送り出す装置。

overflow（あふれ，オーバフロー）　扱い得る量の範囲を超える処理結果を生ずる状態。

例：1.　レジスタの扱い得る数の範囲を超えた計算結果を生ずる状態。

2.　プログラムの大きさが記憶容量を超えた状態。

overhead bit（付加ビット）　誤り制御などの目的で，情報ビットに付加されるビット。

［P］

pack（パック）　データや記憶媒体の特性を利用し，データの原形が復元できるような方法でそのデータを圧縮し，記憶媒体に記憶すること。

packing density（記録密度）　記憶媒体の単位長，単位面積または単位体積当たりの記憶データ量。通常，単位当たりのビット数で表す。recording density ともいう。

paper tape（紙テープ）　一定の幅をもつ紙でできたテープであって，これに一定の規則に従ってせん孔を行うことにより，データを記録することができるもの。

paper tape punch（紙テープせん孔装置）　計算機などからの指令によって，紙テープにデータをせん孔し，記録する装置。

paper tape reader（紙テープ読取装置）　紙テープ上のデータを読み取る装置。

paragraph（段落(C)）　手続き部では，段落名の後ろに幾つかの文を書いたもの。また，見出し部と環境部とでは，段落の見出しの後ろに幾つかの記述項を書いたもの。段落名とは，手続き部の段落を識別するために，段落の先頭に書く手続き名である。段落の見出しとは，見出し部と環境部の段落を識別するために，段落の先頭に書く予約語である。

parallel（並列）（a）　語の各けたを複数個（けた数に等しい個数）の回路で同時に処理すること。

（b）　複数個の装置が同時にデータを処理すること。

parallel operation（並列操作，並行操作）　二つ以上の処理を同時に行う操作。

parallel transmission（並列伝送）　文字，ブロックなどを二つ以上に分割し，複数の通信路で同時に伝送すること。

parameter（パラメタ(A)）　手続きの内容を定義したり，またはそれを使用する際に必要な値の受渡しをしたりするために用いられる文法単位。ALGOL では仮パラメタと実パラメタがある。

参考：FORTRAN では argument という。

parametron（パラメトロン）　共振回路のパラメタ励振現象を利用して，1/2 分周発振を起こさせ，この振動の2種の位相によって2進数字を表示させることにより，記憶や論理演算の機能を行わせる回路素子。

parity bit（奇偶検査ビット，パリティビット）　奇偶検査のために付加されるビット。

parity check（奇偶検査，パリティチェック）　冗長検査の一つであって，2進コードにおいて“1”の個数が奇数または偶数になるように余分のビットを付加し，この2進コードの誤りの有無を検出すること。

pattern recognition（パターン認識）　自動的な手段で形状，輪郭などを識別すること。

peripheral equipment（周辺装置）　データ処理システムにおいて，中央処理装置と主記憶装置を除く装置の総称。

plotter（作図装置，プロッタ）　計算機などからの指令によって，記録面に図形を

描く装置。

point-to-point system（**二地点間方式，ポイントツーポイント方式**）　回線の両端にそれぞれ一つの端末装置などが接続され，その間に他の端末装置などが接続されることのない方式。

polling（**ポーリング**）　個々の端末装置に対して問合せを行い，データの送信を促すこと。

positional notation（**位取り記数法**）　数の表記法の一つであって，各数字の表す値が，それ自身の値とそのけた位置とから決定されるような数字の並びを用いる。

prefetch（**先取り**）　進行中の処理と並行して，必要と思われる命令やデータをあらかじめ読み出すこと。

printer（**印字装置，プリンタ**）　与えられた信号に対応する文字を紙などの媒体に記録する装置。

problem-oriented language（**問題向き言語**）　ある範囲の問題を解くのに向いているプログラム言語。FORTRAN, COBOL などの手順向き言語，GPSS, SIMSCRIPTなどのシミュレーション言語，LISP などのリスト処理言語，情報検索言語など。

procedure（**手順**）　問題解決のためにとる一連の動作。

（**手続き（A, F, C）**）　一連の文の集まりであって，実行することによってある定まった効果を与える文法単位。プログラム中の主要な構文単位の一つである。ALGOLでは，手続き宣言によって定義され，手続き文または関数呼出しによって呼ばれる。特に，式の中で呼ばれる手続きは関数という。FORTRANでは，関数手続きとサブルーチン手続きに分かれる。手続きを引用するプログラム単位の外で定義される手続きを外部手続きといい，文関数と組込み関数を内部手続きという。関数手続きは外部関数及び内部関数（文関数，組込み関数）をいい，サブルーチン手続きは外部サブルーチンともいう。COBOLでは，手続き部の中の一つの段落もしくは節，または機能的に続いた幾つかの段落もしくは節をいう。

procedure division（**手続き部（C）**）　原始プログラムを構成する四つの部の一つであって，最後に書かれる。計算手順が記述される。手続き部は，宣言部分と手続き部分とから成る。これらの部分は節に分かれ，節は更に段落に分かれる。段落は，幾つかの文から成る。

procedure-name（**手続き名（C）**）　文の先頭に付けて文の引用や飛越しの行き先を識別する文字の列。

参考：ALGOLでは label, FORTRANでは statement label という。

procedure oriented language（**手順向き言語**）　手順を具体的なアルゴリズムとして表現できるように作られた問題向き言語。

procedure statement（**手続き文（A）**）　型をもたない形で宣言された手続きの本体を呼び出して実行する文。

procedure subprogram（**手続き副プログラム（F）**）　関数副プログラムまたはサブルーチン副プログラムをいう。

program（**プログラム**）　計算機が受け入れ得る形式になっている一連の命令または命令文。なお，種々の用途に，または何回でも使えるように作られた一連の命令を指す場合がある。routineともいう。

（**プログラム（A）**）　計算機が受け入れ得る形式になっている一連の命令または文。ALGOLでは一つのブロックか複合文である。

参考：FORTRAN では executable program, COBOLでは source program

という。

program language（プログラム言語）　プログラムを表現するために用いられる人工言語。

programmer（プログラマ）　プログラミングをする人。

programming（プログラミング）　プログラムを設計し，書き，テストすること。

program unit（プログラム単位（F））　主プログラムまたは副プログラムをいう。

publication language（発表言語）　基準言語を，印刷，手書き，数学上の慣用などに合わせたもの。これと基準言語との相違点としては，例えば，

　基準言語　$a[i, j]$　$r\uparrow P$　$3_{10}5$

　発表言語　a_{ij}　r^{P}　3×10^{5}

などがある。

punch card（せん孔カード）　一定の規則に従ってせん孔を行うことにより，データを記録することができるカード。

punched card　punch card と同義。

pushdown　last-in first-out と同義。

pushup　first-in first-out と同義。

［Q］

quantity（量（A））　単純変数，配列，名札，スイッチ及び手続きをいう。量には名前が与えられ，その名前は有効範囲をもつ。

［R］

radix（基数，底）　位取り記数法において，あるけたの重みと，1けた上位のけたの重みとの比。

radix notation（基数表記法）　あるけたの重みとその1けた上位のけたの重みとの比が，正の整数であるような位取り記数法。

radix point（小数点）　基数表記法によって表された数の整数部と小数部を分離する位置，またはその位置にある文字。

random access　direct access と同義。

read（読取り，読出し）　記憶媒体からデータを取り出すこと。

read-only storage　fixed storage と同義。

real time processing（実時間処理，リアルタイム処理）　データ処理において，外部事象の発生に応じて要求された時間内にデータを処理する方式。

record（レコード，記録）　情報処理の目的で，一単位として取り扱われる関連した項目，欄または語の集まり。

（記録（F），レコード（C））入出力動作におけるデータ処理の単位。FORTRANでは，記録という。COBOLではレコードという。このほかに，一番包括的なデータ項目をもレコードという。

recording density（記憶密度）　packing density と同義。

record layout（レコード様式，レコードレイアウト）　レコード中のデータや語の配置及び構造のことで，その構成要素の順序と大きさの仕様を含む。

record length（レコード長）　レコードの長さであり，1レコードの中に含まれるバイト数，文字数または語数で表す。

recursive（再帰的）　関数または手続きが再帰的であるとは，その内容を定義する際に自分自身をその定義中で使用することをいう。ALGOLでは，再帰的定義を許しているが，FORTRANとCOBOLでは，禁止している。

redefinition（再定義（F））　定義されていた変数や配列要素が改めて定義されることをいう。

redundancy（冗長度）　データに生ずる誤りを検出または訂正するために，情報を表すのに必要な最小の長さに付加される余分のデータの割合であって，次の式で

表す。

$$1-\frac{k}{n}$$

k：情報を表すのに必要なデータの長さ
n：情報全体を表すデータの長さ

redundancy check（冗長検査）　ビットのあらゆる組合せのうちで，ある規則に合うものだけを使用することにして，誤りの検出または訂正を行う検査方法。

redundant code（冗長符号）　データに生じる誤りを検出または訂正するために，本来の情報を表すのに必要な最小のデータに余分のデータを付加したコードの体系。

reference（引用する（F））　変数や配列要素を引用するという場合と，手続きを引用するという場合がある。変数や配列要素を，プログラム中でその値が使用可能である状態で識別するとき，変数や配列要素を引用するという。手続きを，プログラム中でその実行を要求するという意味で識別するとき，その手続きを引用するという。

reference language（基準言語）　各プログラム言語において，基準となる言語。ハードウェアによる制限，記述の便利さまたは数学上の表記法にとらわれずに，基準としての定義用言語として相互の理解を明確にする立場から定められている。他に発表言語及び金物表現があるが，これらは単に文字の組の一つ一つの外部表現が異なるだけである。

reflected binary code（交番2進コード）　連続する数を2進表現したとき，隣接する数の表現が，互いに一つのけたでだけ異なるように作られた2進コード。

register（レジスタ，置数器）　1ビットまたは複数ビットを保持する装置であって特定の目的に使用され，随時，その内容を利用できるようになっているもの。

relation（比較式（A））　二つの算術式を関係演算子によってつないだ文法単位。その関係が成り立つか成り立たないかに応じて，それぞれ真または偽の値をもつ。ALGOLでは，関係演算子を比較作用素という。比較作用素の両側の算術式は単純算術式でなければならない。

relational condition（比較条件（C））　二つの算術式を関係演算子によってつないだ文法単位。その関係式が成り立つか成り立たないかに応じてそれぞれ真または偽の値をもつ。COBOL では関係演算子を比較演算子という。算術式のほかに文字項目または文字定数の比較も行える。

relational expressions（関係式（F））　二つの算術式を関係演算子によってつないだ文法単位。その関係式が成り立つか成り立たないかに応じて，それぞれ真または偽の値をもつ。

relational operator（比較作用素（A），関係演算子（F），比較演算子（C））　比較式，関係式または比較条件を構成する要素。これらは，次の表のとおりである。

表

比較関係	ALGOL 比較作用素	FORTRAN 関係演算子	COBOL 比較演算子
より小さい	<	.LT.	IS {LESS THAN / <}
等しいかより小さい	≦	.LE.	IS NOT {GREATER THAN / >}
等しい	=	.EQ.	IS {EQUAL TO / =}
等しいかより大きい	≧	.GE.	IS NOT {LESS THAN / <}
より大きい	>	.GT.	IS {GREATER THAN / >}

等しくない	≠	.NE.	IS NOT {EQUAL TO / =}

relative address(相対アドレス)　別に指定される基底アドレスを基準として，相対的に表されたアドレス。

report(報告書(C))　データに対してその編集と制御の切れ目ごとの集計が行われ，指定したページの形で出力されたもの。

request repeat system(再送訂正方式)　データの誤りを検出した場合に，そのデータを再送することにより，誤りを訂正する方式。

reserved word(予約語(C))　ある決まったつづりをもった文字の列であって，原始プログラム中では，定められた文や句の中で，かつ，定められた意味でしか用いることができない。

reset(リセット)　(a)　2値素子を“0”の状態にすること。

(b)　装置，レジスタ，2値素子などを初期状態にすること。

residual error rate(見逃し誤り率)　正しく伝送されなかったエレメント，文字，ブロックなどの数に対する，検出または訂正されなかったものの割合。

ring counter(リングカウンタ，環状計数器)　カウンタの一種であって，2値素子が環状につながれ，通常，その内の1個の素子が他と異なる状態となっていて，入力信号を受けるごとに，この状態が一つ隣りに移動するように構成されたもの。

Roman character(ローマ字)　letter と同義。

ROS　read-only storage の略。

round(丸め)　指定されたけた数に収まるように，与えられた数を近似する数値にすること。丸めには，通常，切上げ，切捨て及び四捨五入(またはこれに相当するもの)がある。

routine(ルーチン)　program と同義。

[S]

scale factor(倍率)　基準化のために用いられる定数。例えば 856，432，−95，−182 という値を −1 と +1 の間に収めるためには，$\frac{1}{1000}$を倍率として選ぶことができる。

scaling(基準化)　値の範囲をあらかじめ定められた限界内に収めるために，定数(倍率)を掛けて値を変えること。

scope(有効範囲(A))　量と名前の対応が，宣言で指定されたとおりに解釈される文と式の集まりをいう。

search(探索)　項目の集まりについて所要の特性をもつものを探すこと。

second level definition(第2階の定義(F))　数値に関する定義において，整数型の変数が添字式または計算型 GO TO 文の中に現れる場合の定義をいう。これは，変数の値が，アドレス，すなわち記憶場所に対応して用いられる場合であり，指標レジスタの最適化を考慮して作られた概念である。

section(節(C))　一つ以上の段落や記述項の組であって，節の見出しで始まる。節の見出しとは，環境部，データ部，手続き部の各節を識別するために，節の先頭に書く語の組である。

selecter channel(選択チャネル，セレクタチャネル)　複数台の入出力装置を接続し，それらを選択的に動作させる入出力チャネル。データ転送は，ブロック単位で行われる。

selecting(セレクティング)　特定の一つ以上の端末装置に対して問合せを行い，データの受信を促すこと。

self correcting system（自己訂正方式） 誤り訂正符号を用いることにより，受信側だけで訂正する方式。

sentence（文(C)） プログラムの主要な単位。原則として書かれた順に実行される。COBOL では，命令(statement)や句を用いてプログラムを書く。文 (sentence)は，幾つかの命令をつないで，終止符と空白とで止めたものである。

命 令 ─無条件命令
─条件命令
─翻訳指示命令

sequence sort と同義。

sequence control counter（逐次制御計数機） instruction address register と同義。

sequential access（順次アクセス） データを記憶装置へ書き込んだり，記憶装置から読み出したりする際に，前回書込みまたは読出しが行われた記憶場所に引き続く記憶場所で，書込みまたは読出しが行われるような方法。

sequential circuit（順序回路） 任意の時点における出力値が，その時点での入力値と回路の内部状態とによって定まる論理回路。その内部状態は直前の入力値と直前の内部状態とによって定まる。

sequential control（逐次制御） あらかじめ指定された命令の系列に従って，計算機を逐次自動的に制御すること。

serial（直列） 語の各けたを一つの回路でつぎつぎに処理すること。

serial operation（直列操作） 時間的に順次に処理する操作。

serial transmission（直列伝送） 信号の各エレメントを時間的な系列として，一つの通信路で1エレメントずつ順次伝送すること。

service bit（サービスビット） チェックビット以外の付加ビット。例えば調歩式に用いるスタートビット，ストップビットなど。

set（セット） (a) 2値素子を“1”の状態にすること。

(b) 装置，レジスタ，2値素子などを指定された状態にすること。

sexadecimal digit hexadecimal digit と同義。

sexadecimal notation hexadecimal notation と同義。

shift（けた送り，けた移動） ビット，バイトまたは文字の並びを右または左に移動させること。

shift register（シフトレジスタ，送りレジスタ） 2値素子の組から成るレジスタの一種であって，シフトパルスが与えられるたびに内容が同時に1けたずつ移動するもの。

side effect（副作用） プログラムにおいて，陽に代入が記述されていないのに，変数や関数の値が変化すること。

signal(信号) 物理現象に関する，時間に依存した値であって，データを伝達するもの。

signal distance hamming distance と同義。

sign bit(符号ビット) 符号位置にあるビットであって，その数の正負を表すもの。

sign digit(符号けた数字) 符号位置にある数字であって，その数の正負を表すもの。

significant condition（有意状態） 信号の各エレメントを特性付ける各々の状態。例えば2値信号を表すための電流のオン（“1”）オフ（“0”）など。

significant instant（有意瞬間） 有意状態が変化する瞬間。例えば電流のオンからオフ，またはオフからオンへの変換点など。

significant interval（有意間隔） 信号の各エレメントの有意状態が維持される時間。

sign position（位置符号） 数の正負を表すけた位置。通常は，数字の並びの一端にある。

simplex（単向） データを回線上のあらかじめ決められた一方向にだけ伝送することができる方式。

simulation（シミュレーション） 物理的システムまたは抽象的システムの行動の幾つかの特徴を，別のシステムの行動によって表現すること。

例：1.　計算機の動作によって物理現象あるいは経済現象など，表現をすること。

2.　ある計算機の動作を別の計算機の動作で表現すること。

simulator（シミュレータ） シミュレーションを行う装置，データ処理システムまたはプログラム。

single address instruction　one address instruction と同義。

software（ソフトウェア） データ処理システムの動作に関するプログラム，手順及び関連する書類。ハードウェア（hardware）の対照語。

sort（分類） 項目をキーに従って，必要な順序に並べること。例えば数の大きさの順，アルファベット順，年代順に分類するなど。

sorter（分類機） せん孔カード，OCR用紙，MICR用紙などを分類する装置。

source language（原始言語） 翻訳処理の入力に用いられる言語。

source program（原始プログラム） 原始言語によって表現されたプログラム。
（原始プログラム（C）） 計算機が受け入れ得る形式になっている一連の命令または文。COBOL では，見出し部，環境部，データ部及び手続き部から成る。

space character（間隔文字） 単語を分けるための図形文字であって，通常，何も表示されない図形で表される。

special character（特殊文字） 数字，英字，仮名，漢字，間隔文字のいずれでもない図形文字。

例：，。＋ － ￥ 〃（濁点）など。

special control character（特殊制御文字） 伝送制御文字，書式制御文字，装置制御文字，情報分離文字のいずれでもない制御文字。

special purpose computer（専用計算機） 限られた分野の問題だけを処理するように設計されている計算機。

specification statment（宣言文(F)） データの特性や並び方のための情報を指定する文であって，DIMENSION 文，COMMON 文，EQUIVALENCE 文，EXTERNAL 文及び型宣言文がある。

specification subprogram（初期値設定副プログラム(F)） FORTRANの用語。副プログラムの一つであって，BLOCK DATA 文を先頭に置き，型宣言文，DATA 文，COMMON 文などから構成され，END 行で終わる。名前付き共通ブロックの要素に初期値を与えるのに使われる。

stacker（スタッカ） カードなどを扱う装置において，処理された後のカードなどを受け入れる部分。ホッパと対比される。

standard data format（標準データ形式（C）） データ部において，データの性質を記述するときの概念であって，無限の幅と長さをもった印刷ページ上のデータ形式を想定して，データの性質や特性を表現するもの。

standard procedure（標準手続き(A)） プログラマが定義しないで使うことので

きる手続き。入出力は，すべて標準手続きを用いて行われる。

standard function（標準関数(A)）　プログラマが定義しないで使うことのできる関数。

start-stop system（調歩式）　エレメント群(例えば，文字に対応する個々の符号)に対して，スタート信号及びストップ信号をそれぞれ先行及び後続させるような同期の一形式。

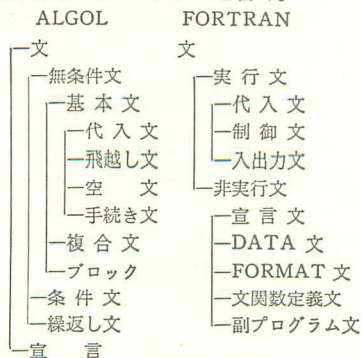
statement（命令文）　プログラム言語における表現形式の一単位であって，それによって演算その他の動作を記述したり指定したりするもの。

（文（A, F））　プログラムの主要な構成単位。原則として書かれた順に実行される。ALGOL では，文（statement）や宣言などを用いてプログラムを書く。文には，無条件文，条件文及び繰返し文がある。FORTRAN では，実行文と非実行文を用いてプログラムを書く。

ALGOL

- 文
  - 無条件文
    - 基 本 文
      - 代 入 文
      - 飛越し文
      - 空　　文
      - 手続き文
    - 複 合 文
    - ブロック
  - 条 件 文
  - 繰返し文
- 宣　　言

FORTRAN

- 文
  - 実 行 文
    - 代 入 文
    - 制 御 文
    - 入出力文
  - 非実行文
    - 宣 言 文
    - DATA 文
    - FORMAT 文
    - 文関数定義文
    - 副プログラム文

（命令(C)）　sentence と同義。

statement label（文の番号(F)）　文の先頭に付けて，文の引用や飛越しの行き先を識別する文字の列。ALGOL では label, COBOL では procedure-name という。

storage（記憶装置）　計算機の構成要素の一つであって，計算処理に必要なデータを記憶する装置。

storage capacity（記憶容量）　記憶装置に蓄え得るデータの量。通常，ビット数，バイト数，文字数または言語で表す。

storage location（記憶場所）　内部記憶装置の中で，アドレスによって指定された場所。

store（記憶（場所））（a）　データを後で取り出せるような方法によって，一時期または永久的に保存すること。

（b）　storage と同義。

stored program computer（プログラム記憶式計算機）　内部的に記憶された命令によって制御される計算機であって，プログラムを読み取り，記憶し，引き続きその命令を実行できるもの。

subprogram（サブプログラム）　subroutine と同義。

（副プログラム(F)）　実行可能プログラムの構成単位の一つであって，関数副プログラム，サブルーチン副プログラム及び初期値設定副プログラムに分かれる。なお，関数副プログラム及びサブルーチン副プログラムを手続き副プログラムともいう。

subprogram statement（副プログラム文(F)）　プログラム単位の種類を指定する文であって，FUNCTION 文，SUBROUTINE 文及び BLOCKDATA 文がある。

subroutine（サブルーチン）　プログラムの一部分であって，問題の一部を計算するための，それ自身でまとまった命令の系列。

subroutine subprogram（サブルーチン副プログラム(F)）　副プログラムの一つであって，SUBROUTINE 文を先頭とする幾つかの文によって定義される外部サブルーチン。CALL 文によって引用

され，実行の効果を引き数や共通ブロックに与えることができる。

subscript（添字(A, F, C)）　配列中の要素の位置を識別するために，配列名の後ろに添えるものであって，算術式の並びを括弧でくくったもの。FORTRAN 及び COBOL では，この算術式の構文には強い制限がある。

substracter（サブストラクタ）　(a)　二つの数の差を作る回路。

(b)　3個の入力端子と2個の出力端子とをもち，出力信号が入力信号に対し，次の表の関係にある回路。

X → ［減算器］ → D
Y → 　　　　　 → B
B* →

X：被減数
Y：減数
B*：下位での借り
D：差
B：上位からの借り

表

入力			出力	
X	Y	B^*	D	B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	0
1	1	0	0	0
0	0	1	1	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	1

symbolic address（記号アドレス）　プログラミングに便利なように，記号で表されたアドレス。

symbolic name（英文名（F））　変数，配列，手続きなどを識別するのに用いる文字の列。

synchronization（同期）　データ伝送において，データ通信速度を合わせるなどの約束に基づいて，各エレメントまたはエレメント群の時間関係を，送受信相互間で一致させること。

備考：同期は，次のように分類される。

同期 { 同期式 / 非同期式(調歩式) }

synchronous computer（同期式計算機）　個々の事象や基本的動作の実行が，刻時パルスによって制御され，これと歩調を合わせて進行するように作られた計算機。

synchronous system（同期式）　どの二つの有意瞬間をとっても，その間隔が単位間隔の整数倍である同期の一形式。

system（システム）　人間，機械，方式などの集まりであって，特定の諸機能を果たすために組織されているもの。

［T］

table（表，テーブル）　一つ以上の引き数によって，一意に識別される項目の配列。

（表(C)）　COBOL では，表といい，添字または指標の付いたデータ名で表の要素を識別する。

tag（タグ）　flag と同義。

target language（目的言語）　翻訳処理の出力に用いられる言語。

target program（目的プログラム）　原始言語から翻訳され目的言語になっているプログラム。

terminal equipment（端末装置）　通信回線を介してデータを入力し，または出力するために用いられる装置。

time sharing（時分割，タイムシェアリング）　一つの装置を二つ以上の目的のために時間を細分して使用し，見掛け上同時に動作が行われるようにすること。

tracer（追跡プログラム）　被検査プログラムが正しく働いているかどうかを調べるために，実行過程において必要な情報を出力させるプログラム。

tracing program　tracer と同義。

track（トラック）　磁気ドラム，磁気ディスク，磁気テープなどの表面に一連のデータを蓄え，1個のヘッドで書込みまたは読取りができる連続した線状の部分。

transaction file（トランザクションファイル，発生ファイル）　新規に発生した一時的なデータを含むファイル。このファイルは適用業務に応じて所要のマスタファイルと突き合わされて処理される。

transfer（転送）　ある所からデータを送り，他の所でそのデータを受け取ること。例えば周辺装置と中央処理装置との間で，入出力チャネルを通してデータを転送するなど。

transfer time（転送時間）　記憶装置からまたは記憶装置へ，データの転送が開始されてから転送が完了するまでの時間。

transmission control character（伝送制御文字）　通信回線におけるデータの伝送を制御し，または容易にするために用いられる制御文字。

transmit　transfer と同義。

trap（トラップ）　interruption と同義。

trigger circuit（トリガ回路）　少なくとも一つの安定状態を含む多くの安定状態または不安定状態をもち，適切なパルスの印加によって目的とする状態への転移が引き起こされる回路。

two address instruction（2アドレス命令）　命令の形式の一種であって，アドレスを2個含むもの。

type（型（A, F））　データを数学的なまたは表現上の性質により分類したもの。ALGOL では，型といい，整数型，実数型及び論理型がある。FORTRAN では，型といい，整数型，実数型，倍精度実数型，複素数型，論理型及び文字型がある。
参考：COBOL では category という。

［U］

unconditional jump instruction（無条件飛越命令）　次に実行する命令を，指定したアドレスから取ることを要求する命令。

unconditional statement（無条件文(A)）　論理式または条件に制御されることなく定められた動作を行う文法単位。
参考：COBOL では imperative statement という。

undefined（不定（F））　変数や配列要素が定義されていないとき，その変数や配列要素は不定であるという。

underflow（下位けたあふれ，アンダフロー）　絶対値が扱い得る数の範囲に満たない小さい計算結果を生ずる状態。浮動小数点演算において許された範囲に入らない負の指数が生じた場合，下位けたあふれが起こる。

undetected error rate　residual error rate と同義。

unpack（アンパック）　パックされたデータを元の形のデータに直すこと。

［V］

variable（変数(A, F)）　プログラム中の最も基本的な文法単位。名前によって識別され，これに値を与えることができる。変数は文中で値が代入されたり，その値が引用されたりする。ALGOLでは，変数といい，単純変数と添字付き変数とがある。FORTRAN では，変数という。
参考：COBOL では，data item という。

variable word length（可変語長）　記憶装置の基本語長が可変である形式。

verifier（検孔機）　せん孔カードなどに与えたデータの誤りの有無を手動により検査する装置。

vertical check（垂直検査）　媒体に記録された2進コードの検査の方式であって，媒体の運動方向に対し垂直な方向のビットについて奇偶検査などを行うこと。

volume（ボリューム）　一つの単位として，

取り付けたり取り外したりできる記憶媒体。例えば個々の磁気テープまたはディスクパック。

[W]

word（語，単語）　一単位として取り扱われる一連のビット，バイトまたは文字の列。

word length（語長）　語の長さであり，1語の中に含まれるビット数，バイト数または文字数で表す。

write（書込み）　記憶媒体にデータを入れること。

write enable ring（書込み許可リング）　磁気テープにデータを記録するとき，リールに取り付けるリングであって，これを取り外すことにより書込みを禁止し，ファイルを保護するためのもの。file protection ring ともいう。

[Z]

zero suppression（ゼロ抑制）　数値の中の不必要なゼロを表示しないこと。

備考：不必要なゼロには，数値の整数部の最上位の非ゼロ数字の左側のゼロと小数部の最下位の非ゼロ数字の右側のゼロが含まれる。

zone（ゾーン）　ビットの組合せにより文字を表す場合，数字以外の文字を表す目的に使用される幾つかの特定のビットが置かれる場所。例えば，80けたのせん孔カードでは，1けたに12個のせん孔位置があるが(図参照)，数字は0から9までのせん孔位置によって表示されるのに対し，数字以外の文字は上3個のせん孔位置と1から9までのせん孔位置との組合せによって表示される。

和 文 索 引

<第Ⅰ部，第Ⅱ部用>

ア

アーキテクチャ……70
アクセス……123
アセンブラ……113
アプリケーション……18
アールピージー……113
アンダーフロー……131
暗に……36

イ

1次的……127
入れ子……32
インタプリタ……64
インデックスシーケンシャルファイル……28
インテリジェントターミナル……77
インプット／アウトプット……20
インプリシット……36

エ

エクスプリシット……36
エーピーエル……113
演　算……41
エントリ……23

オ

オーバーフロー……131
オペランド……41
オペレーション……41
オペレーティングシステム……21

カ

開始ブロック……49
外部装置……27
拡張化2進10進コード……43
仮想記憶装置……139
仮想小数点……12
環境部……14
緩衝記憶装置……129
関　数……23
関数サブプログラム……23

キ

キーイン……51
機械から独立……6
機械語……113
機械に依存しない……5
機械向き……19
基底アドレス……138
境　界……24
キーワード……46
キーワードステートメント……46

ク・ケ

区域A……16
区域B……16
空白ステートメント……46
空白文……47
グループ……48
原始プログラム……20

コ

構成単位……27
高水準計算機用言語……4
高密度集積回路……76
国際標準化機構……3
固定小数点リテラル……12
コード入力システム……91
コレーティングシーケンス……43
コンセプト……70
コントロールブレーク……111
コンパイラ……6, 20

サ

索引順次編成……122

索引順次編成ファイル…… 28
サポート…… 72

シ

ジェネレート……111
識別名…… 37
シーケンシャルファイル…… 28
システムハウス…… 85
実 行…… 52
実効アドレス……102
指標レジスタ……102
時分割処理……135
シミュレーション……126
集積回路…… 76, 82, 101
順次編成……123
順次編成ファイル…… 23
照合順序…… 43
状 態…… 52
省略時解釈…… 36
診 断…… 30

ス

数字リテラル…… 12
スケーリング……131
ステートメント…… 32, 46
ステートメント識別語…… 47
ステートメント本体…… 46
ストリング演算…… 34
スーパーバイザ…… 64

セ

制御の切れ目……111
静 的……33, 39, 65
精 度……23, 131
絶対アドレス……138
絶対番地……138
戦略的計画……117

ソ

相対アドレス……138
相対番地……138
装 置…… 27
属 性…… 35, 37
ソースプログラム…… 20
ソフトウェア……128

タ

第1世代…… 97
第2世代…… 97
第3世代…… 97
第3･5世代…… 97
第4世代…… 97
大規模集積回路…… 82, 93
タイムシェアリングシステム……108
ダイレクトファイル…… 28
単 位…… 27

チ

チップメーカー…… 85
知能端末…… 77
致命度…… 31
中央処理装置……102, 128, 135
超 LSI…… 82
直接編成ファイル…… 28

テ・ト

定 数…… 10
ディレクトリ…… 79
デコーダ……103
データシステム言語会議…… 3
データ部…… 14
データベース管理システム……73, 149
手続き……141
手続き部…… 14
デバッギング…… 19
電子郵便……107
動 的……33, 38, 65
閉じたサブルーチン……140

ナ・ニ・ネ

内部装置…… 27
2次的……127
2進法……134
日本語情報処理…… 91, 95
入出力…… 20
ネスト式…… 32

ハ

ハイレベルランゲージ……………… 6
バウンダリ………………………… 24
パソコン…………………………… 93
パーソナルコンピュータ………… 93
バーチャルメモリ………………139
バッテリインディケータ………… 59
番　地……………………………103

ヒ

ビギンブロック…………………… 49
ビット……………………………134
ビット列…………………………… 34
評　価……………………………… 41
開いたサブルーチン……………140

フ

部…………………………………… 14
部見出し…………………………… 14
プリプロセッサ…………………111
プロシージャ……………………… 32
プロシージャブロック…………… 48
ブロック…………………………… 48
プロトコル………………………… 73
フロントエンド…………………149
文………………………………32,46
分散処理…………………………… 71

ヘ・ホ

米国規格協会……………………… 3
ベーシック………………………113
ベースアドレス…………………139
ペンタッチ方式…………………… 91
補　数……………………………132

マ・ミ

マイクロコンピュータ……77,84,85,89
マイクロプロセッサ……………… 76
マシン……………………………… 19
マシンランゲージ………………113
待ち行列…………………………109
マルチストロークシステム……… 90
丸　め……………………………131
見出し部…………………………… 14

メ

明に………………………………… 36
命　令…………………………103,104
命令実行段階……………………103
メ　ガ……………………………… 61

モ

模　型……………………………126
文字セット……………………9,42,44
文字列……………………………… 34
モデル……………………………126
モード……………………………… 52
問題向き…………………………… 19

ユ・ヨ

ユーザ仕様書……………………125
予約語……………………………… 10

ラ・リ

ランダムファイル………………… 28
乱編成…………………………122,123
乱編成ファイル…………………… 28
リテラル…………………10,11,12
利用技術…………………………128

ル・レ

ルックアップ方式………………… 91
レベルインディケータ…………… 17
レベル番号………………………… 17
レベル標識………………………… 17

ワ

ワードプロセッシング…………107
ワーニング………………………… 31
割当てステートメント…………… 46
割当て文…………………………… 46
割込み……………………………108

英 文 索 引

<第Ⅰ部, 第Ⅱ部用>

A

absolute address ……… *138*
Account Receivable ……… *62*
address ……… *102, 103*
alphameric ……… *43*
alphanumeric ……… *43*
AMDAHL ……… *136*
ANSI ……… *2, 3*
APL ……… *113*
application ……… *18*
architecture ……… *70*
Area A ……… *16*
Area B ……… *16*
ASSEMBLER ……… *113*
Assignment ……… *45*
AUTOMATIC ……… *38*

B

backspace ……… *28*
base address ……… *138*
BASED ……… *38*
BASIC ……… *55, 112, 113*
battery indicator ……… *59*
binary ……… *105*
binary digit ……… *134*
binary notation ……… *134*
bit ……… *134*
bit per inch ……… *101*
bit string ……… *34*
blink ……… *67*
boundary ……… *24*
BPI ……… *101*
branch ……… *104*
buffer ……… *129*

C

cathode ray tube ……… *101*
central processing unit ……… *101, 128, 135*
character string ……… *34*
chip ……… *85*
closed subroutine ……… *140*
COBOL ……… *112, 129*
CODASYL ……… *2, 3*
collating sequence ……… *43*
communication ……… *144*
compiler ……… *6, 20*
complement ……… *132*
concept ……… *70*
contextual ……… *36*
control break ……… *111*
CONTROLLED ……… *38*
conversion ……… *64*
core memory ……… *100*
costing ……… *69*
CPU ……… *101, 102, 128, 135*
CRT ……… *101*

D

DAM ……… *28*
Data Base ……… *73*
Data Base Management System ……… *73*
DATA DIVISION ……… *14, 15*
DBMS ……… *73, 149*
debugging ……… *19*
decoder ……… *103*
diagnostic ……… *30*
direct access method ……… *28*
direct file ……… *28*
display ……… *51*
distributed processing ……… *71*
division header ……… *14*
dot ……… *95*
dynamic ……… *33, 65*

E

effective address ……102
electronic mail ……107
end-of-file record ……27
entry ……23
ENVIRONMENT DIVISION ……14, 15
evaluation ……41
execution ……52
execution cycle ……103
explicit ……36
external or internal unit ……27

F・G・H

fetch cycle ……103
fixed-point numeric literal ……12
fioating-point numeric literal ……12
FORTRAN ……112
generate ……111
generator ……111
high-level computer language ……6
home budgeting ……69

I

IC ……76, 82
IDENTIFICATION DIVISION ……14, 15
identifier(s) ……37
implicit ……36
indexed sequential access method ……28
indexed sequential file ……28
indexed sequential organization ……122
index register ……102
ink-jet ……107
input/output ……20
instruction ……104
integrated circuit ……76, 101
intelligent terminal ……77
interface ……26
interpretive ……64
interrupt ……108
inventory control ……62, 69
inventory management ……62
ISAM ……28
ISO ……3

J・K

JECC ……83
JEIDA ……83
JICST ……83
JIPDEC ……83
KDD ……83
key in ……51

L

Large Scale Integration ……76
laser-xerography ……107
l.p.m. ……61
LSI ……76, 82, 93

M

machine ……19
machine-independent ……5
machine language ……113
machine-oriented ……7, 19
magnetic tape ……101
main memory ……100
main storage ……100
margin A ……16
margin B ……16
micro computer ……77, 84
MITI ……83
model ……127
modify ……105
MPT ……83
multiprogramming system ……140

N

NEC ……83
nest ……82
nonnumeric literal ……12
NTT ……83

O

on-line ……129
open-ended ……138
open subroutine ……140

operand ………… *41*
operating system ………… *21*
operation ………… *41*
overflow ………… *131*
O.S. ………… *21*

P・Q

payroll ………… *69*
personal computer ………… *93*
precision ………… *23, 131*
problem-oriented ………… *5, 19*
procedure ………… *141*
PROCEDURE DIVISION ………… *14, 15*
protocol ………… *73*
queue ………… *109*

R

relative address ………… *138*
routine ………… *26*
round-off ………… *131*
RPG ………… *113*
run ………… *52*

S

SAM ………… *28*
scaling ………… *131*
screen ………… *51*
sequential access method ………… *28*
sequential file ………… *28*
severity ………… *31*
single touch ………… *67*
software ………… *128*
source program ………… *20*
statement identifier ………… *47*
static ………… *33, 38, 65*
stock control ………… *69*
stock management ………… *69*
storage allocation ………… *141*
strategic planning ………… *117*
supervisor ………… *64*
swapped out ………… *109*
system house ………… *85*

T

tab ………… *59*
table ………… *23*
ten-key ………… *67*
thruput ………… *39*
time sharing system ………… *108*
to place ………… *57*
to put ………… *57*
transistor ………… *101*
transition ………… *64*
TSS ………… *108, 135*
turnkey program ………… *63*

U・V

underflow ………… *131*
unit ………… *27*
virtual memory ………… *139*
VLSI ………… *82, 86, 87*

<編者紹介>

日 高 哲 郎（ひだか　てつろう）

1972年電気通信大学電気通信学部電子工学科卒業．自動制御を専攻・(財)日本情報処理開発協会を経て現在(株)エス・イー・アシスト代表取締役

著書

アセンブラ演習，共立出版

COBOL 8週間，共立出版

佐 藤 文 博（さとう　ふみひろ）

1974年早稲田大学教育学部教育学科卒業．教育工学を専攻．現在(財)日本情報処理開発協会中央情報教育研究所調査企画部調査企画課課長

中央大学経済学部兼任講師

解説 コンピュータ英語

1983 年 3 月 25 日　初版 1 刷発行
1993 年 3 月 15 日　初版19刷発行

編者　日 高 哲 郎
　　　佐 藤 文 博

発行　共立出版株式会社／南條正男

東京都文京区小日向 4—6—19
電話東京 (03) 3947 局 2511 番（代表）
郵便番号 112／振替口座 東京 1-57035 番

印刷　科学図書印刷
製本　協 栄 製 本

社団法人
自然科学書協会
会員

検印廃止
NDC 007.6
ISBN 4-320-02227-0

Printed in Japan